



TUGAS AKHIR - TE 141599

**PERANCANGAN KONTROLER PID DENGAN
GENETIC ALGORITHM UNTUK *DIRECT TORQUE*
CONTROL DALAM PENGENDALIAN MOTOR
INDUKSI 3 FASA**

Adeta Gita Ahmadani
NRP. 2215 105 047

Dosen Pembimbing
Ir. Joko Susila, MT.

DEPARTEMENT TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017

Halaman ini sengaja dikosongkan



FINAL PROJECT - TE 141599

***DESIGN OF PID CONTROLLER WITH GENETIC
ALGORITHM FOR DIRECT TORQUE CONTROL IN 3
PHASE INDUCTION MOTOR CONTROLLING***

Adeta Gita Ahmadani
NRP 2215 105 047

Supervisor
Ir. Joko Susila, MT.

***ELECTRICAL ENGINEERING DEPARTMENT
Fakulty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya
2017***

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Perancangan Kontroler PID dengan *Genetic Algorithm* untuk *Direct Torque Control* dalam Pengendalian Motor Induksi 3 fasa”** adalah benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2017

Adeta Gita Ahmadani
NRP. 2215105047

Halaman ini sengaja dikosongkan

**PERANCANGAN KONTROLER PID DENGAN GENETIC
ALGORITHM UNTUK DIRECT TORQUE CONTROL DALAM
PENGENDALIAN MOTOR INDUKSI 3 FASA**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik
Pada

Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I,

Ir. Joko Susila, M.T.
NIP. 196606061991021001



Halaman ini sengaja dikosongkan

PERANCANGAN KONTROLER PID DENGAN *GENETIC ALGORITHM* UNTUK *DIRECT TORQUE CONTROL* DALAM PENGENDALIAN MOTOR INDUKSI 3 FASA

Nama : Adeta Gita Ahmadani

Dosen Pembimbing : Ir. Joko Susila M.T.

ABSTRAK

Dalam dunia Industri penggerak elektrik banyak digunakan dalam sebuah proses produksi terutama yang banyak digunakan adalah motor Induksi. Motor induksi banyak digunakan dikarenakan mempunyai konstruksi yang kuat, kecepatan putaran yang konstan terhadap variasi beban, harganya murah, perawatannya yang mudah, dan lain – lain. Namun, pengendalian motor induksi tidak mudah karena parameternya yang berubah – ubah membuat karakteristik dari motor induksi tidak linear. Sehingga diperlukan suatu studi tentang pengendalian motor induksi sehingga dapat mengatasi kesulitan yang ada. Terdapat beberapa cara dalam mengendalikan motor induksi salah satunya adalah metode *Direct Torque Control* yang memiliki struktur pengaturan yang sederhana dan tidak banyak perhitungan. Pada Tugas Akhir ini dilakukan perancangan dengan menggunakan *Direct Torque Control* dengan kontrol PID untuk mengatur kecepatan motor pada kondisi *steady state* maupun kondisi *transien* pada saat terjadi perubahan beban pada motor. Parameter pada PID dapat diestimasi dengan menggunakan metode kontrol cerdas *Genetic Algorithm*. Selain PID sebagai pembandingan dilakukan juga pembandingan dengan menggunakan *Fuzzy Logic Controller*(FLC). Ketiga metode kontrol dilakukan perancangan dan disimulasikan pada MATLAB. Hasil yang didapatkan adalah Kontroler PID-GA mampu mengatasi penurunan kecepatan lebih sedikit pada saat terjadi perubahan beban yang diberikan yaitu sebesar 0,8 rps dibandingkan dengan PID dan FLC dengan penurunan sebesar 10 rps. *Ripple* torsi yang didapatkan juga lebih kecil dengan nilai 1,78 Nm dibandingkan dengan PID sebesar 2,2 Nm dan FLC sebesar 2,0 Nm.

Kata Kunci : Motor Induksi, *Direct Torque Control*(DTC), *Genetic-Algorithm*, PID

Halaman ini sengaja dikosongkan

DESIGN OF PID CONTROLLER WITH GENETIC ALGORITHM FOR DIRECT TORQUE CONTROL IN 3 PHASE INDUCTION MOTOR CONTROLLING

Name : Adeta Gita Ahmadani
Supervisor : Ir. Joko Susila M.T.

ABSTRACT

Industrial Sector often using electric machine for production proses commonly used is induction motor. Induction Motor often used because of its strong construction, constanly radian along variation of load change, cheap price, easy maintenance, etc. However, the controlling of induction motor is not easy because the parameter is always changed makes the characteristic is non-linear. Accordingly, needed some study about controlling induction motor so that the problem can be resolved. There are few ways in induction motor controlling one of those is Direct Torque Control that have simply structure controlling and no many calculation. The final project is design controller using Direct Torque Control with PID for controlling the radiant velocity on steady state or transien state when the load change is happen. PID Parameter can be estimated using Genetic-Algorithm that can give better performane . PID Controller and Fuzzy Logic Controller(FLC) designed for the comparison for PID-GA. Both of three designed and simulated in MATLAB. The Result is PID-GA can give better performance that when system is weighted the decrease of speed can be reduced with value is 0.8 rps. Its better than PID and FLC that give decrease of speed is 10 rps. PID-GA can give better result in ripple with value is 1.78 Nm than PID with 2.2 Nm and FLC with 2.0 Nm.

Keyword : Induction Motor, Direct Torque Control(DTC), Genetic-Algorithm, PID

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur Alhamdulillah penulis panjatkan kehadiran ALLAH SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya, sehingga penelitian dan penulisan tugas akhir dengan judul **“Perancangan Kontroler PID dengan *Genetic Algorithm* untuk *Direct Torque Control* dalam Pengendalian Motor Induksi 3 fasa”** ini dapat diselesaikan dengan baik.

Penulis menyadari bahwa selama proses penyusunan tugas akhir ini mengalami kendala-kendala, namun berkat nasihat, bimbingan, bantuan dari berbagai pihak dan berkah dari Allah SWT sehingga semua kendala yang ada dapat diatasi oleh penulis. Dalam kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Kedua Orang tua tercinta serta seluruh anggota keluarga yang telah banyak memberikan perhatian, dukungan, semangat doa kepada penulis.
2. Bapak Ir. Joko Susila M.T. selaku dosen pembimbing yang telah memberi banyak ilmu dan arahan, semoga bapak dan ibu sehat selalu dan diberi keberkahan ilmu.
3. Seluruh dosen Jurusan Teknik Elektro-ITS, yang telah banyak memberikan ilmu serta motivasi dalam memahami setiap ilmu yang dipelajari.
4. Seluruh teman-teman LJ Elektro terutama rekan seperjuangan Teknik Sistem Pengaturan yang telah banyak memberikan masukan, bantuan dan motivasi, sukses selalu untuk kita semua teman-teman.

Dalam menyusun tugas akhir ini, penulis berharap semoga nanti tugas akhir ini akan bermanfaat terhadap semua pihak dan penulis mengharapkan saran dan kritik terhadap buku ini.

Surabaya, Juni 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	i
ABSTRAK	vii
<i>ABSTRACT</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Tugas Akhir dan Manfaat	2
1.5 Metodologi	2
1.6 Sistematika Penulisan	3
1.7 Relevansi	4
BAB 2 DASAR TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Motor Induksi	6
2.2.1 Sistem Kelistrikan Motor Induksi	8
2.2.2 Prinsip Kerja Motor Induksi	9
2.2.3 Pemodelan Motor Induksi 3 fasa	10
2.3 <i>Direct Torque Control</i> (DTC)	17
2.3.1 Komparator Histerisis	20
2.3.2 <i>Estimator</i> dan Sektor Fluks Stator	21
2.3.3 <i>Switching Table</i>	21
2.4 Kontroler <i>Proporsional Integral Derivative</i> (PID)	23
2.5 Sistem Pengaturan Cerdas	25
2.6 <i>Genetic Algorithm</i> (GA)	25
2.6.1 Seleksi	29
2.6.2 <i>Crossover</i>	30
2.6.3 Mutasi	31
2.7 Kontroler PID- <i>Genetic Algorithm</i> (PID-GA)	32

BAB 3 PERANCANGAN SISTEM	35
3.1 Perancangan Blok Diagram.....	35
3.2 Perancangan Model Motor Induksi 3 Fasa.....	36
3.3 Perancangan <i>Direct Torque Control</i> (DTC) Motor Induksi..	39
3.3.1 Perancangan <i>Estimator</i> dan Sektor Sudut Fluks.....	39
3.3.2 Perancangan Blok Komparator.....	41
3.3.3 Perancangan Blok <i>Switching Table</i>	42
3.4 Perancangan Kontroler <i>Fuzzy Logic Controller</i> (FLC)- DTC	43
3.5 Perancangan Kontroler PID-GA.....	43
 BAB 4 SIMULASI DAN HASIL	45
4.1 Simulasi Model Motor Induksi.....	45
4.2 Simulasi dan Pengujian <i>Direct Torque Control</i> (DTC) dengan Kontroler PID.....	46
4.2.1 Simulasi dan Pengujian Tanpa Beban.....	46
4.2.2 Pengujian dengan Menggunakan Perubahan Beban	48
4.3 Simulasi dan Pengujian <i>Direct Torque Control</i> (DTC) dengan <i>Fuzzy Logic Controller</i> (FLC)	49
4.4 Simulasi dan Pengujian <i>Direct Torque Control</i> (DTC) dengan Kontroler PID-GA.....	51
4.4.1 Proses <i>Learning</i> Algoritma Genetika.....	51
4.4.2 Pengujian Kontroler PID-GA	59
 BAB 5 PENUTUP.....	65
5.1 Kesimpulan.....	65
5.2 Saran.....	65
DAFTAR PUSTAKA.....	67
LAMPIRAN.....	69
RIWAYAT HIDUP PENULIS	77

TABLE OF CONTENTS

TITLE PAGE	i
STATEMENT OF AUTHENTICATION	i
ABSTRACT(BAHASA)	vii
ABSTRACT(ENGLISH).....	ix
PREFACE	xi
TABLE OF CONTENTS.....	xiii
LIST OF FIGURES	xv
LIST OF TABLE	xix
 CHAPTER 1 INTRODUCTION	 1
1.1 Background.....	1
1.2 Problem Formulation	2
1.3 Problem Constrains	2
1.4 Purpose and Benefit of The Final Project	2
1.5 Methodology.....	2
1.6 Writing Systematics	3
1.7 Relevance	4
 BAB 2 BASIC THEORY	 5
2.1 Literature Review	5
2.2 Induction Motor	6
2.2.1 Electricity System of Induction Motor	8
2.2.2 Work Principle of Induction Motor	9
2.2.3 Modelling of 3 Phase Induction Motor	10
2.3 Direct Torque Control(DTC).....	17
2.3.1 Hysterisis Comparator.....	20
2.3.2 Estimator and Stator Fluks Sector.....	21
2.3.3 Switching Table	21
2.4 Proporsional Integral Derivative(PID) Controller	23
2.5 Artificial Inteligence Control System	25
2.6 Genetic Algorithm(GA)	25
2.6.1 Selecton	29
2.6.2 Crossover	30
2.6.3 Mutation	31
2.7 PID-Genetic Algorithm(PID-GA) Controller.....	32

BAB 3 SYSTEM DESIGN	35
3.1 Diagram Block Design	35
3.2 Design of 3 Phase Induction Motor Model	36
3.3 Design of Direct Torque Control(DTC) Induction Motor ...	39
3.3.1 Design of Estimator and Angle of Fluks Sector	39
3.3.2 Design of Comparator Block	41
3.3.3 Design of Switching Table Block	42
3.4 Design of Fuzzy Logic Controller(FLC)-DTC.....	43
3.5 Design of PID-GA Controller.....	43
 BAB 4 SIMULATION AND RESULT	 45
4.1 Simulation of Induction Motor Model	45
4.2 Simulation and Testing of Direct Torque Control (DTC) with PID Controller	46
4.2.1 Unweighted Simulation and Testing	46
4.2.2 Weighthed Change Simulation	48
4.3 Simulation and Testing of Direct Torque Control (DTC) with Fuzzy Logic Controller(FLC)	49
4.4 Simulation and Testing of Direct Torque Control (DTC) with PID-GA Controller	51
4.4.1 LearningProcess Genetics Algorithm.....	51
4.4.2 Testing of PID-GAController	59
 BAB 5 FINALE	 65
5.1 Conclusion	65
5.2 Suggestion	65
BIBLIOGRAPHY	67
APENDIX	69
BIOGRAPHY	77

DAFTAR GAMBAR

Gambar 2.1 Motor Induksi.....	5
Gambar 2.2 Perbandingan Kurva Karakteristik Motor	6
Gambar 2.3 Bagian-bagian Motor Induksi	6
Gambar 2.4 Rotor Tipe <i>Wound</i> dan Rotor Tipe <i>Squirrel Cage</i>	7
Gambar 2.5 Sistem 3 fasa	8
Gambar 2.6 <i>Coupling Effect</i> pada Belitan Stator dan Belitan Rotor Kumparan dua Fasa	10
Gambar 2.7 Rangkaian Ekuivalen Motor Induksi Tiga Fasa pada Sumbu d-q.....	11
Gambar 2.8 Representasi Koordinat 3 Fasa dan 2 Fasa	15
Gambar 2.9 Representasi 3 Fasa dalam Koordinat 2 Fasa (α, β)	16
Gambar 2.10 <i>Transformasi Park</i>	17
Gambar 2.11 Diagram Blok <i>Direct Torque Control</i>	18
Gambar 2.12 Pembagian Ruang Vektor Tegangan Bidang dq	20
Gambar 2.13 Kontroler Histerisis Fluks dan Kontroler Histerisis Torsi	21
Gambar 2.14 Sektor Tegangan <i>Inverter</i> dan Sektor <i>Switching</i> Fluks Stator pada DTC Konvensional dan DTC Modifikasi	23
Gambar 2.15 Kontroler PID.....	23
Gambar 2.16 Bagian individu pada <i>Genetic Algorithm</i>	26
Gambar 2.17 Proses <i>Genetic Algorithm</i>	29
Gambar 2.18 Proses <i>Tournament Selection</i> GA	30
Gambar 2.19 <i>Single Point Crossover</i>	31
Gambar 2.20 <i>Arithmetic Crossover</i>	31
Gambar 2.21 Mutasi GA.....	32
Gambar 2.22 Kontroler PID-GA.....	33
Gambar 3.1 Diagram Blok Simulasi	35
Gambar 3.2 Model <i>Simulink</i> Motor Induksi	38
Gambar 3.3 Perancangan <i>Estimator</i> DTC	39
Gambar 3.4 Transformasi Tegangan dari abc ke dq	40
Gambar 3.5 Perancangan Sektor Sudut Fluks.....	41
Gambar 3.6 Blok Komparator Torsi dan Blok Komparator Fluks..	42
Gambar 3.7 Blok <i>Switching Table</i>	42
Gambar 3.8 Blok Diagram <i>Fuzzy Logic Controller</i> untuk DTC.....	43
Gambar 3.9 Representasi Kromosom Individu	44

Gambar 4.1 Respon Kecepatan Rotor	45
Gambar 4.2 Respon Torsi Rotor	46
Gambar 4.3 Respon Kecepatan Kontroler PID Tanpa Beban.....	47
Gambar 4.4 Respon Torsi Kontroler PID Tanpa Beban.....	47
Gambar 4.5 Respon Kontroler PID Terbeban.....	48
Gambar 4.6 Respon Torsi Kontroler PID Terbeban.....	48
Gambar 4.7 Respon Kecepatan Kontroler FLC tanpa Beban	49
Gambar 4.8 Respon Torsi Kontroler FLC tanpa Beban	49
Gambar 4.9 Respon Kecepatan Kontroler FLC Terbeban.....	50
Gambar 4.10 Respon Torsi Kontroler FLC Terbeban.....	50
Gambar 4.11 Nilai <i>Fitness</i> Maksimal Probabilitas Crossover=0,8 Probabilitas Mutasi =0,05	52
Gambar 4.12 Nilai <i>Fitness</i> Rata-rata Probabilitas Crossover=0,8 Probabilitas Mutasi =0,05	52
Gambar 4.13 Nilai <i>Fitness</i> Maksimal Probabilitas Crossover=0,8 dan Probabilitas Mutasi =0,6.....	53
Gambar 4.14 Nilai <i>Fitness</i> Rata-rata Probabilitas Crossover=0,8 dan Probabilitas Mutasi =0,6.....	53
Gambar 4.15 Nilai <i>Fitness</i> Maksimal Probabilitas Crossover=0,3 dan Probabilitas Mutasi =0,02	54
Gambar 4.16 Nilai <i>Fitness</i> Rata-rata Probabilitas Crossover=0,3 dan Probabilitas Mutasi =0,02	54
Gambar 4.17 Nilai <i>Fitness</i> Maksimal Probabilitas Crossover=0,1 dan Probabilitas Mutasi =0,1	55
Gambar 4.18 Nilai <i>Fitness</i> Rata-rata Probabilitas Crossover=0,1 dan Probabilitas Mutasi =0,1	55
Gambar 4.19 Nilai <i>Fitness</i> Maksimal Probabilitas Crossover=0,01 dan Probabilitas Mutasi =0,2.....	56
Gambar 4.20 Nilai <i>Fitness</i> Rata-rata Probabilitas Crossover=0,01 dan Probabilitas Mutasi =0,2.....	56
Gambar 4.21 Nilai <i>Fitness</i> Maksimal Probabilitas Crossover=0,3 dan Probabilitas Mutasi =0,02	57
Gambar 4.22 Nilai <i>Fitness</i> Rata-rata Probabilitas Crossover=0,3 dan Probabilitas Mutasi =0,02	57
Gambar 4.23 Nilai <i>Fitness</i> Maksimal Probabilitas Crossover=0,05 dan Probabilitas Mutasi =0,8	58
Gambar 4.24 Nilai <i>Fitness</i> Rata-rata Probabilitas Crossover=0,05 dan Probabilitas Mutasi =0,8	58

Gambar 4.25 Respon Kecepatan Kontroler PID-GA Tanpa Beban	59
Gambar 4.26 Respon Torsi Kontroler PID-GA Tanpa Beban	60
Gambar 4.27 Respon Kecepatan Kontroler PID-GA Terbeban.....	61
Gambar 4.28 Respon Torsi Kontroler PID-GA Terbeban.....	61
Gambar 4.29 Perbandingan Respon Kecepatan PID, FLC dan PID-GA	62
Gambar 4.30 Perbandingan Respon Torsi PID, FLC dan PID- GA.....	62

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Pengaruh Tiap-Tiap Vektor Tegangan terhadap Nilai Fluks Stator dan Torsi Elektromagnetik.....	20
Tabel 2.2 <i>Switching Table</i> Vektor Tegangan <i>Inverter</i>	22
Tabel 2.3 Perbandingan K_p , K_i , dan K_d	24
Tabel 4.1 Hasil Identifikasi sistem	51
Tabel 4.2 Perbandingan Karakteristik Respon PID dan PID-GA	63

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Penggerak elektrik banyak digunakan dalam aplikasi kehidupan sehari-hari baik industri maupun bidang lainnya. Dalam dunia Industri penggerak elektrik banyak digunakan dalam sebuah proses produksi terutama yang banyak digunakan adalah motor Induksi. Motor induksi banyak digunakan dalam suatu proses produksi karena konstruksi yang kuat, kecepatan putaran yang konstan terhadap variasi beban, harganya murah, perawatannya yang mudah, dan lain – lain. Namun, pengendalian motor induksi tidak mudah karena parameternya yang berubah – ubah membuat karakteristik dari motor induksi tidak *linear*.

Pada pengaturan motor Induksi terdapat beberapa cara salah satunya adalah metode pengaturan motor induksi dengan metode vektor kontrol yang berdasarkan hubungan pada kondisi dinamik. Pada metode vektor kontrol tidak hanya mengatur besar dan frekuensi kecepatan angular tetapi juga mengatur posisi dari vektor ruang tegangan, arus dan fluks stator. Metode vektor kontrol yang banyak digunakan adalah *Field-Oriented Control (FOC)* yang dapat mengatur motor induksi seperti motor DC eksitasi terpisah. Akan tetapi, metode FOC mempunyai beberapa kekurangan yaitu membutuhkan transformasi koordinat, pengaturan arus, sensitif terhadap perubahan parameter dan sistem menjadi lebih kompleks. Untuk mengatasi masalah tersebut digunakan metode *Direct Torque Control (DTC)* yang struktur pengaturannya lebih sederhana dan tidak banyak perhitungan. Metode DTC memungkinkan secara langsung mengatur kondisi *switching* pada inverter dengan *Space Vector Modulation*.

Metode DTC dapat dimodifikasi dengan PID untuk mengatur kecepatan motor pada kondisi *steady state* maupun kondisi *transien* pada saat terjadi perubahan beban pada motor. Parameter pada PID dapat diestimasi dengan menggunakan metode kontrol cerdas *Genetic-Algorithm* sehingga didapatkan pengendalian yang lebih baik. Tambahan kontroler cerdas *Genetic-Algorithm* diharapkan mampu memberikan hasil yang baik dalam pengaturan motor induksi

1.2 Perumusan Masalah

Perumusan masalah yang dibahas dalam Tugas Akhir ini adalah pengaturan kecepatan motor induksi tiga fasa dengan *Direct Torque Control (DTC)* dengan PID berbasis *Genetic-Algorithm* untuk memperbaiki respon kecepatan dari motor induksi tiga fasa sehingga kecepatannya sesuai dengan referensi yang diberikan.

1.3 Batasan Masalah

Ada beberapa batasan yang terdapat dalam pembahasan tugas akhir ini, yaitu sebagai berikut:

1. *Plant* yang dikontrol merupakan motor induksi 3 fasa dalam bentuk model d-q. Pemodelan motor induksi sendiri didapatkan dari buku referensi.
2. Ruang lingkup yang dibahas yaitu permasalahan respon kecepatan pada saat *steady state*, kemampuan sistem dalam mengatasi perubahan beban dan meminimalkan fluktuasi *ripple* fluks dan torsi
3. Optimasi parameter *PID* dilakukan secara *off-line* menggunakan Algoritma Genetika

1.4 Tujuan Tugas Akhir dan Manfaat

Tujuan dari Penelitian ini adalah untuk merancang kontrol motor induksi menggunakan *Direct Torque Control* dengan PID berbasis *Genetic-Algorithm* untuk pengaturan kecepatan motor induksi tiga fasa sehingga dapat memperbaiki respon kecepatan dari motor induksi tiga fasa saat terjadi perubahan.

1.5 Metodologi

Metodologi yang dilakukan dalam pengerjaan Tugas akhir ini terdiri dari beberapa tahap, yaitu :

1. Studi Literatur
Studi literatur dilakukan untuk memperoleh informasi mengenai motor induksi tiga fasa melalui buku teks, jurnal, artikel, internet dan lain-lain.
2. Pemodelan Sistem
Pada tahap ini dibuat pemodelan motor induksi tiga fasa serta *Direct Torque Control (DTC)*

3. Desain Kontroler
Pada tahap ini dibuat struktur kontroler PID-GA untuk pengaturan kecepatan motor induksi tiga fasa.
4. Simulasi
Pemodelan motor induksi dan hasil desain kontroler disimulasikan dengan menggunakan perangkat lunak MATLAB.
5. Penyusunan Buku Tugas Akhir
Penyusunan buku Tugas Akhir meliputi pendahuluan, teori penunjang, perancangan sistem, simulasi dan analisa sistem serta penutup.

1.6 Sistematika Penulisan

Pada Tugas Akhir ini sistematika penulisan dibagi menjadi lima bab, yaitu :

BAB 1 : Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2 : Teori Penunjang

Bab ini membahas tinjauan pustaka yang membantupenelitian, di antaranya konsep motor induksi tiga fasa, teori pemodelan dinamik motor induksi, teori *Direct Torque Control* (DTC), dan teori kontroler PID-GA.

BAB 3 : Perancangan Sistem

Pada bab ini dijelaskan mengenai perancangan model dinamik motor induksi tiga fasa dan *Direct Torque Control* (DTC) serta algoritma kontroler PID-GA.

BAB 4 : Simulasi dan Analisa Sistem

Bab ini memuat hasil simulasi kontroler pada sistem dan analisa sistem.

BAB 5 : Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan dari hasil yang telah diperoleh

1.7 Relevansi

Hasil dari Tugas Akhir ini diharapkan dapat menjadi referensi dan pengembangan pada bidang penelitian mesin listrik dan kontrol cerdas dengan Algoritma Genetika

BAB 2

DASAR TEORI

2.1 Tinjauan Pustaka[1],[2]

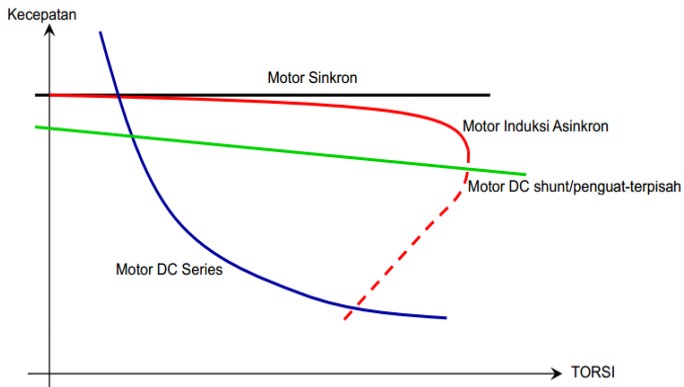
Pada dunia industri jenis penggerak yang paling banyak digunakan adalah motor induksi. Motor induksi mempunyai konstruksi yang kokoh, efisiensi daya yang tinggi, dan terutama perawatannya yang mudah. Hal itulah yang menyebabkan motor induksi banyak digunakan dari pada jenis motor lainnya motor induksi ditunjukkan pada Gambar 2.1.



Gambar 2.1 Motor Induksi

Pada saat ini 80% dari motor penggerak pada dunia industri merupakan motor induksi dan akan terus bertambah seiring perkembangan waktu. Sehingga studi untuk motor induksi menjadi hal yang penting sebagai salah satu penyokong kemajuan dunia industri. Selain pada dunia industri motor induksi juga banyak digunakan pada bidang transportasi dengan perkembangan kendaraan tenaga listrik yang semakin pesat.

Namun berbeda dengan motor penggerak jenis lain motor lain motor induksi sulit dikontrol dikarenakan pengaturan torsi dan fluks menjadi satu sehingga menyebabkan karakteristiknya *nonlinear*. Oleh karena itu penelitian tentang pengontrolan motor induksi menjadi hal yang menarik untuk dilakukan. Perbandingan kurva karakteristik motor ditunjukkan pada Gambar 2.2.

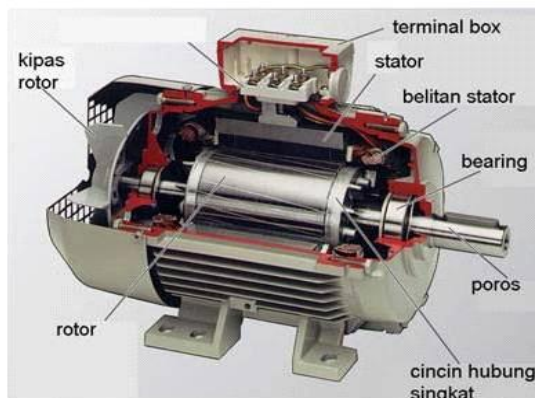


Gambar 2.2 Perbandingan Kurva Karakteristik Motor

Dengan melihat Gambar 2.2 dapat dilihat bahwa jika dibandingkan motor lain motor induksi asinkron mempunyai karakteristik yang *nonlinear*.

2.2 Motor Induksi[3]

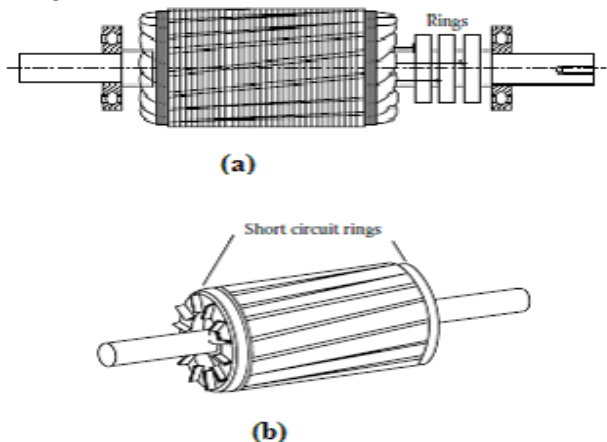
Pada Gambar 2.3 ditunjukkan bagian-bagian dari motor induksi. Motor induksi terdiri dari 2 komponen utama yaitu stator dan rotor.



Gambar 2.3 Bagian-bagian Motor Induksi

Stator adalah bagian motor induksi yang diam pada saat beroperasi sedangkan rotor adalah bagian yang bergerak. Stator adalah rangkaian lilitan yang dibentuk kumparan dan disuplai oleh tegangan AC tiga fasa yang masing-masing berbeda fasa 120 derajat. Pada stator dengan kutub 1 terdapat 3 kumparan yang membentuk sudut 120 derajat yang masing-masing kumparan mendapatkan 1 fasa dari sumber AC.

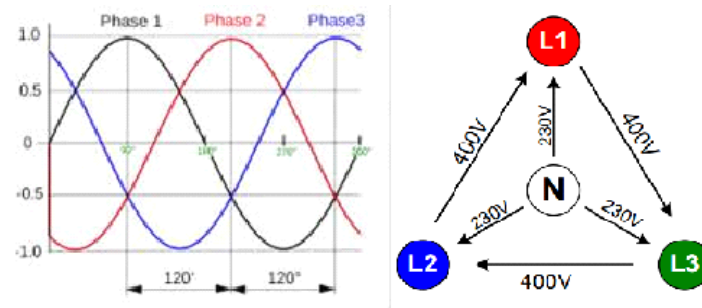
Rotor adalah bagian motor induksi yang berputar untuk dikopel dengan beban. Rotor terdiri dari rangkain kumparan 3 fasa yang dihubung singkat disusun secara internal(tipe *wound* dengan slip ring) atau secara eksternal melalui sikat arang. Pada rotor tipe *wound* rangkaian kumparan pada rotor serupa dengan rangkaian kumparan pada stator yang mendapatkan suplai 3 fasa yang suplai pada masing-masing rangkaian didapatkan dari 3 slip ring yang terpasang pada poros stator. Kebanyakan kumparan rotor tersambung secara bintang dengan 3 fasanya tersambung pada slip ring. *Slip* ring terhubung oleh sikat arang yang terpasang di stator yang menyentuh slip ring secara terus menerus. Rotor yang paling banyak digunakan adalah rangkaian rotor dengan kumparan internal yang sering disebut dengan *squirrel cage* atau rotor sangkar tupai. Rotor sangkar tupai terbentuk dari sejumlah batang yang dimiringkan dan terhubung singkat pada ujung-ujungnya. Gambar 2.4 menunjukkan tipe rotor *wound* dan tipe rotor *squirrel cage*.



Gambar 2.4 Rotor Tipe *Wound* dan Rotor Tipe *Squirrel Cage*

2.2.1 Sistem Kelistrikan Motor Induksi[2]

Motor induksi asinkron mendapatkan energi dari suplai listrik agar bekerja. Untuk mengetahui prinsip kerja dari motor induksi 3 fasa ada baiknya perlu dimengerti sistem kelistrikan pada motor induksi. Sistem jaringan 3 fasa terdiri dari 3 konduktor fasa, satu konduktor netral dan satu konduktor ground. Di Indonesia sistem 3 fasa bekerja pada frekuensi 50 Hz. Gelombang dan hubungan setiap konduktor sistem 3 fasa ditunjukkan pada Gambar 2.5.



Gambar 2.5 Sistem 3 Fasa

Pada Gambar 2.5 dapat dilihat bahwa selisih antar gelombang fasa adalah 120 derajat. Kemudian dengan menggunakan hubungan bintang maka selisih antara fasa dan netral(N) adalah 230 Volt sedangkan selisih antara fasa dan fasa(U) adalah 400 Volt. Sehingga diketahui jika tegangan fasa 1 dan netral (L1) digambarkan sebagai gelombang sinus dengan persamaan matematis

$$V_1 = V_{max} \sin \omega t \quad (2.1)$$

Dengan : $\omega = 2\pi f$

Maka bisa diketahui tegangan antara fasa dan netral untuk L2 dan L3

$$V_2 = V_{max} \sin \left(\omega t - \frac{2\pi}{3} \right) \quad (2.2)$$

$$V_3 = V_{max} \sin \left(\omega t - \frac{4\pi}{3} \right) \quad (2.3)$$

Daya pada sistem jaringan 3 fasa adalah total dari jumlah daya pada masing-masing fasa. Pada masing-masing fasa besar daya

tegangan antara fasa dan netral dikalikan dengan arus yang mengalir pada beban. Arus yang mengalir pada beban bergantung pada sifat beban itu sendiri. Sifat beban mempengaruhi pergeseran fasa dari gelombang yang disebut dengan $\cos\phi$ daya dan bisa dibentuk dalam Persamaan 2.4

$$P = U \times I\sqrt{3}\cos\phi \quad (2.4)$$

Untuk daya beban yang bersifat resistif maka pergeseran fasanya 0 derajat sehingga

$$P = U \times I\sqrt{3} \quad (2.5)$$

2.2.2 Prinsip Kerja Motor Induksi[4]

Prinsip kerja motor induksi berkaitan erat dengan Hukum Faraday dan Hukum Lorentz. Hukum Faraday menyatakan bahwa jika konduktor memotong suatu medan magnet konstan maka pada konduktor tersebut terjadi GGL atau gaya gerak listrik sehingga pada konduktor tersebut timbul tegangan. Hukum Lorentz menyatakan jika suatu konduktor teraliri listrik memotong sebuah medan magnet maka pada konduktor tersebut terjadi sebuah gaya sehingga membuat konduktor tersebut bergerak.

Prinsip kerja motor induksi dapat dijelaskan dengan urutan sebagai berikut :

1. Stator yang terdiri dari 3 kumparan konduktor masing-masing teraliri arus listrik 1 fasa dari suplai 3 fasa yang diberikan. Antara satu fasa dengan fasa disebelahnya tertinggal 120 derajat sehingga pada kumparan stator menimbulkan medan magnet yang berputar. Medan magnet yang berputar tersebut mempunyai kecepatan yang disebut dengan kecepatan sinkron(N_s) yang mempunyai besar :

$$N_s = \frac{60.f}{p} \quad (2.6)$$

Dengan N_s = Kecepatan Sinkron(rpm)
 p = Jumlah kutub dari kumparan stator
 f = Frekuensi stator pada motor induksi(Hertz)

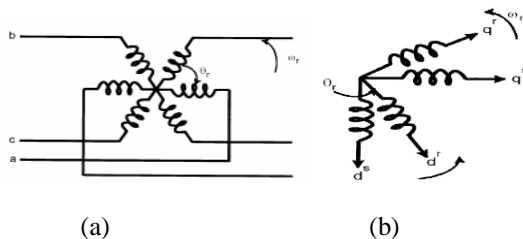
2. Medan magnet yang berputar tersebut kemudian memotong konduktor pada rotor. Sesuai Hukum Faraday maka pada konduktor timbul gaya gerak listrik sehingga konduktor yang dihubungkan singkat pada kedua ujungnya teraliri arus listrik.
3. Arus listrik bersimpangan dengan medan magnet yang dihasilkan oleh stator. Sesuai Hukum Lorentz arus listrik tersebut menghasilkan gaya mekanik atau bisa disebut torsi elektromagnetik yang membuat konduktor bergerak sehingga rotor berputar.
4. Kecepatan putar rotor yang dihasilkan oleh proses elektromagnetik mempunyai perbedaan kecepatan dengan kecepatan sinkron medan magnet stator. Beda kecepatan tersebut disebut dengan *slip* dimana

$$S = \frac{\Delta N}{N_s} = \frac{\Delta \omega}{\omega_s} = \frac{N_s - N_r}{N_s} = \frac{\omega_s - \omega_r}{\omega_s} \quad (2.7)$$

di mana,
 ω_s = Kecepatan angular stator(rad/s)
 N_r = Kecepatan rotor(rpm)
 ω_r = Kecepatan elektrik rotor(rad/s)

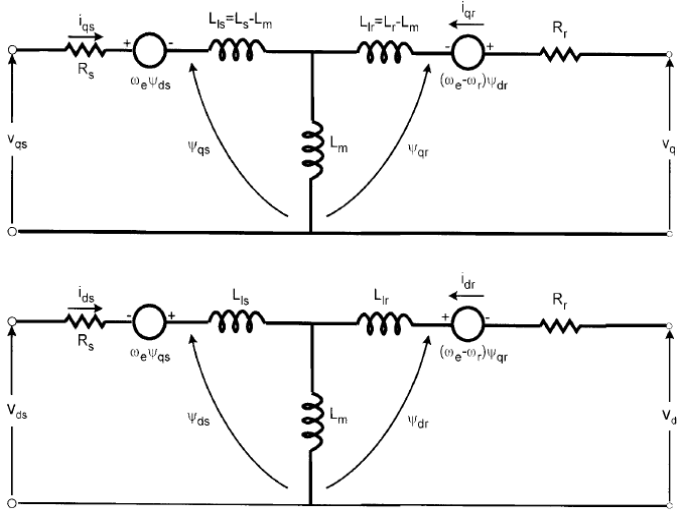
2.2.3 Pemodelan Motor Induksi 3 fasa[1],[4]

Pemodelan motor induksi 3 fasa dilakukan dengan menganalisa pemodelan motor induksi 2 fasa pada bidang *direct*(d) dan *quadrature*(q) terlebih dahulu karena kompleksitasnya. Analisa motor induksi 3 fasa sulit dilakukan karena terdapat *coupling effect* pada belitan stator dan belitan rotor. Pendekatan ini bisa digunakan untuk pemodelan untuk motor induksi n-fasa. *Coupling effect* ditunjukkan pada Gambar 2.6.



Gambar 2.6 *Coupling Effect* pada Belitan Stator dan Belitan Rotor Kumparan Dua Fasa

Kumparan dua fasa pada Gambar 2.6(b) diperlukan representasi rangkaian d_s - q_s dan d_r - q_r dalam frame d_e - q_e yang berputar secara sinkron. Rangkaian motor induksi tiga fasa dalam sumbu d-q dapat dilihat pada Gambar 2.7.



Gambar 2.7 Rangkaian Ekuivalen Motor Induksi Tiga Fasa pada Sumbu dq-

Berdasarkan rangkaian pada Gambar 2.7 maka dapat dituliskan persamaan tegangan pada stator dan rotor dinyatakan dengan Persamaan 2.8.

$$\begin{aligned}
 V_{qs} &= R_s \cdot i_{qs} + \omega \cdot \lambda_{ds} + S \cdot \lambda_{qs} \\
 V_{ds} &= R_s \cdot i_{ds} - \omega \cdot \lambda_{qs} + S \cdot \lambda_{ds} \\
 V_{qr} &= R_r \cdot i_{qr} + (\omega - \omega_r) \cdot \lambda_{dr} + S \cdot \lambda_{qr} \\
 V_{dr} &= R_r \cdot i_{dr} + (\omega - \omega_r) \cdot \lambda_{qr} + S \cdot \lambda_{dr}
 \end{aligned}
 \tag{2.8}$$

dimana $S = \frac{d}{dt}$.

Persamaan fluks pada masing-masing kumparan dapat dinyatakan dengan Persamaan 2.9.

$$\begin{aligned}
\lambda_{qs} &= L_s \cdot i_{qs} + L_m \cdot i_{qr} \\
\lambda_{ds} &= L_s \cdot i_{ds} + L_m \cdot i_{dr} \\
\lambda_{qr} &= L_r \cdot i_{qr} + L_m \cdot i_{qs} \\
\lambda_{dr} &= L_r \cdot i_{dr} + L_m \cdot i_{ds}
\end{aligned} \tag{2.9}$$

sehingga apabila Persamaan 2.8 dimasukkan ke Persamaan 2.9 maka dalam bentuk matriks dapat dinyatakan menjadi Persamaan 2.10.

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s + SL_s & \omega \cdot L_s & SL_m & \omega \cdot L_m \\ -\omega \cdot L_s & R_s + SL_s & -\omega \cdot L_m & SL_m \\ SL_m & (\omega - \omega_r) L_m & R_r + SL_r & (\omega - \omega_r) L_r \\ -(\omega - \omega_r) L_m & SL_m & -(\omega - \omega_r) L_r & R_r + SL_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \tag{2.10}$$

dengan,

$$\begin{aligned}
L_s &= L_{is} + L_m \\
L_r &= L_{ir} + L_m
\end{aligned} \tag{2.11}$$

dimana,

- V_{qs}, V_{ds} = tegangan stator pada sumbu dq (Volt)
- V_{qr}, V_{dr} = tegangan rotor pada sumbu dq (Volt)
- i_{qs}, i_{ds} = arus stator pada sumbu dq (Ampere)
- i_{qr}, i_{dr} = arus rotor pada sumbu dq (Ampere)
- $\lambda_{qs}, \lambda_{ds}$ = fluks stator pada sumbu dq (Webber)
- $\lambda_{qr}, \lambda_{dr}$ = fluks rotor pada sumbu dq (Webber)
- R_s = tahanan stator (Ohm)
- R_r = tahanan rotor (Ohm)
- L_s = induktansi stator (Henry)
- L_r = induktansi diri rotor (Henry)
- L_m = induktansi mutual (Henry)

Persamaan 2.10 dilakukan penyederhanaan agar memudahkan dalam pembuatan simulasi dalam *software* yang dipakai. Proses penyederhanaan Persamaan 2.10 dapat dilihat pada Persamaan 2.12

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s + pL_s & \omega \cdot L_s & pL_m & \omega \cdot L_m \\ -\omega \cdot L_s & R_s + pL_s & -\omega \cdot L_m & pL_m \\ pL_m & (\omega - \omega_r)L_m & R_r + pL_r & (\omega - \omega_r)L_r \\ -(\omega - \omega_r)L_m & pL_m & -(\omega - \omega_r)L_r & R_r + pL_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (2.12)$$

dilakukan pemisahan antara variabel yang mengandung turunan, sehingga bentuk persamaannya menjadi Persamaan 2.13

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s & \omega \cdot L_s & 0 & \omega \cdot L_m \\ -\omega \cdot L_s & R_s & -\omega \cdot L_m & 0 \\ 0 & (\omega - \omega_r)L_m & R_r & (\omega - \omega_r)L_r \\ -(\omega - \omega_r)L_m & 0 & -(\omega - \omega_r)L_r & R_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} + \begin{bmatrix} pL_s & 0 & pL_m & 0 \\ 0 & pL_s & 0 & pL_m \\ pL_m & 0 & pL_r & 0 \\ 0 & pL_m & 0 & pL_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (2.13)$$

dalam bentuk lain, Persamaan 2.13 dapat dituliskan menjadi Persamaan 2.14

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \begin{bmatrix} R_s & \omega \cdot L_s & 0 & \omega \cdot L_m \\ -\omega \cdot L_s & R_s & -\omega \cdot L_m & 0 \\ 0 & (\omega - \omega_r)L_m & R_r & (\omega - \omega_r)L_r \\ -(\omega - \omega_r)L_m & 0 & -(\omega - \omega_r)L_r & R_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} + \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & pL_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (2.14)$$

$$\text{misal, } \mathbf{P} = \begin{bmatrix} R_s & \omega \cdot L_s & 0 & \omega \cdot L_m \\ -\omega \cdot L_s & R_s & -\omega \cdot L_m & 0 \\ 0 & (\omega - \omega_r)L_m & R_r & (\omega - \omega_r)L_r \\ -(\omega - \omega_r)L_m & 0 & -(\omega - \omega_r)L_r & R_r \end{bmatrix}$$

$$\text{dan } \mathbf{Q} = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & pL_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix}.$$

Sehingga Persamaan 2.14 dapat dibentuk menjadi lebih sederhana menjadi Persamaan 2.15

$$\begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} = \mathbf{P} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} + \mathbf{Q} \frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (2.15)$$

Apabila yang ingin dicari nilai arus, maka Persamaan 2.15 dapat dituliskan menjadi Persamaan 2.16

$$\frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} = \mathbf{Q}^{-1} \begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} - \mathbf{Q}^{-1} \mathbf{P} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \quad (2.16)$$

Dari Persamaan 2.16, bisa didapatkan nilai arus dan fluks sehingga bisa dicari nilai torsi elektromagnetik yang dibangkitkan dengan menggunakan Persamaan 2.17

$$T_{em} = \frac{3}{2} \left(\frac{P}{2} \right) L_m \left(i_{qs} i_{dr} - i_{ds} i_{qr} \right) \quad (2.17)$$

dimana,

T_{em} = Torsi Elektromagnetik yang dibangkitkan (Nm)

P = Jumlah Pasang Kutub / *poles*

Kecepatan putar dari motor induksi sendiri bisa ditentukan oleh perbedaan torsi elektromagnetik dan torsi beban (termasuk torsi gesekan) yang berkerja pada inersia gabungan beban dan motor (J)[6].

$$\frac{d}{dt} \omega_{mech} = \frac{T_{em} - T_L}{J} \quad (2.18)$$

$$\omega_r = \frac{P}{2} \omega_{mech} \quad (2.19)$$

dimana,

ω_{mech} = Kecepatan Mekanik Rotor (rad/s)

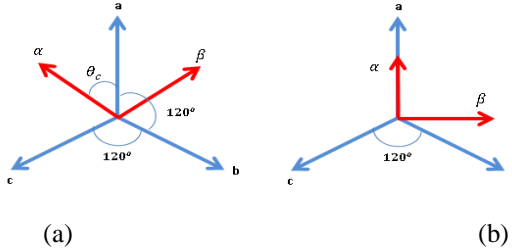
ω_r = Kecepatan Elektrik Rotor (rad/s)

T_L = Torsi Beban (Nm)

J = Momen Inersia (Kg.m^2)

$$\begin{bmatrix} i_{qs} \\ i_{ds} \\ i_o \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos\theta_c & \cos\left(\theta_c - \frac{2\pi}{3}\right) & \cos\left(\theta_c - \frac{2\pi}{3}\right) \\ \sin\theta_c & \sin\left(\theta_c - \frac{2\pi}{3}\right) & \sin\left(\theta_c - \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} \quad (2.21)$$

Untuk merubah koordinat 3 fasa menjadi 2 fasa terdapat dua proses perubahan yaitu merubah koordinat 3 fasa menjadi koordinat 2 fasa pada keadaan stasioner terlebih dahulu yang disebut dengan *stator references frame* (α, β) dan dilanjutkan merubah *stator references frame* dalam keadaan stasioner ke keadaan berputar atau *synchronously rotating frame* (d,q). Perubahan koordinat 3 fasa ke dalam 2 fasa dengan *stator references frame* (α, β) dapat digambarkan seperti Gambar 2.9(a).



Gambar 2.9 Representasi 3 Fasa dalam Koordinat 2 Fasa (α, β)

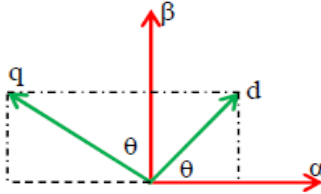
Jika nilai θ_c dibuat nol maka representasi 3 fasa ke 2 fasa dalam bidang stasioner berubah seperti ditunjukkan pada Gambar 2.9(b). Sehingga didapatkan matriks transformasi seperti pada Persamaan 2.22.

$$T = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2}\sqrt{3} & -\frac{1}{2}\sqrt{3} \end{bmatrix} \quad (2.22)$$

Persamaan 2.22 juga dapat digunakan sebagai variabel tegangan maupun variabel arus. Sehingga jika variabel i diganti dengan x maka Persamaan 2.21 bisa dibuat menjadi Persamaan 2.23

$$\begin{bmatrix} x_{\alpha s} \\ x_{\beta s} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2}\sqrt{3} & -\frac{1}{2}\sqrt{3} \end{bmatrix} \begin{bmatrix} x_{as} \\ x_{bs} \\ x_{cs} \end{bmatrix} \quad (2.23)$$

Transformasi diatas disebut dengan *Transformasi Clarke*. Untuk mendapatkan persamaan dalam keadaan *synchronously rotating frame* (dq) dapat digunakan *Transformasi Park*. *Transformasi Park* ditunjukkan oleh Gambar 2.10



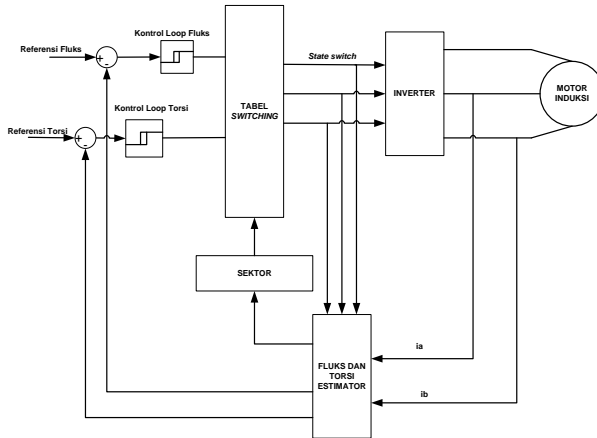
Gambar 2.10 *Transformasi Park*

Persamaan untuk *Transformasi Park* ditunjukkan pada Persamaan 2.24.

$$\begin{bmatrix} X_d \\ X_q \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X_{\alpha s} \\ X_{\beta s} \end{bmatrix} \quad (2.24)$$

2.3 Direct Torque Control(DTC)[4],[5]

Direct Torque Control pertama kali diperkenalkan oleh Takahashi dan Noguchi pada tahun 1986. *Direct Torque Control* adalah metode kontrol motor induksi dengan mengontrol secara langsung perubahan fluks dan torsiya berdasarkan estimasi fluks stator yang didapatkan. *Direct Torque Control* juga dapat mengestimasi *speed* atau kecepatan yang didapatkan sehingga bisa diterapkan tanpa memerlukan sensor kecepatan. Metode ini dapat menghasilkan pengontrolan yang cepat dengan struktur yang sederhana sehingga populer digunakan dalam dunia industri. Diagram blok dari DTC ditunjukkan pada Gambar 2.11.



Gambar 2.11 Diagram Blok *Direct Torque Control*

Melihat Gambar 2.11 dapat ditunjukkan bahwa DTC terdiri dari 2 loop kontrol yang berhubungan dengan fluks dan torsi yang akan dikontrol. Pada masing-masing fluks terdapat referensi yang akan dibandingkan dengan nilai dari *estimator*. *Switching Table* diberikan untuk menerjemahkan aksi kontrol yang akan dilakukan. Hasil dari terjemahan *Switching Table* berupa vektor tegangan yang akan dimasukkan pada *inverter*.

Beberapa kelebihan DTC antara lain sebagai berikut:

- Performansi bagus dan bisa dilakukan tanpa menggunakan sensor kecepatan
- Tidak ada umpan balik kontrol arus
- Tidak ada algoritma *Pulse Width Modulation(PWM)* yang diterapkan.
- Tidak ada transformasi vektor seperti pada vektor kontrol
- Pemrosesan sinyal umpan balik hampir sama dengan fluks stator pada kontrol vektor.
- Kontrol batas histerisis menghasilkan riak(*ripple*) fluks, torsi, dan frekuensi *switching* yang tidak konstan(seperti batas histerisis kontrol arus)

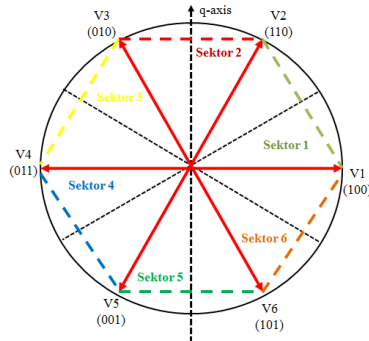
Prinsip dasar DTC dapat diformulasikan sebagai berikut:

- a. Fluks stator merupakan perubahan atau penambahan nilai tegangan EMF. Besarnya nilai fluks stator sangat bergantung pada tegangan stator.
- b. Torsi elektromagnetik yang dibangkitkan berbanding lurus dengan nilai sinusoidal dari sudut antara vektor fluks stator dan rotor.
- c. Perubahan nilai fluks rotor terhadap tegangan stator lebih lambat dibandingkan fluks stator.

Berdasarkan prinsip dasar DTC yang dijelaskan di atas, besarnya nilai fluks stator dan torsi elektromagnetik yang dibangkitkan dapat secara langsung dikontrol dengan memilih secara tepat vektor tegangan stator, yaitu pemilihan *state inverter* secara berurutan dengan spesifikasi berikut:

- a. Vektor tegangan bukan nol yang tidak sejajar dengan vektor fluks stator tidak melebihi $\pm 90^\circ$ menyebabkan fluks naik.
- b. Vektor tegangan bukan nol yang tidak sejajar dengan vektor fluks stator melebihi $\pm 90^\circ$ menyebabkan fluks turun.
- c. *State* nol, V_0 dan V_7 (pada saat durasi pendek) tidak berpengaruh pada vektor fluks stator yang berarti motor berhenti bergerak.
- d. Torsi elektromagnetik yang dibangkitkan dapat dikontrol dengan memilih *state inverter* yang vektor fluks statornya dipercepat, diberhentikan atau diperlambat.

Bidang dq dibagi menjadi enam sektor yang masing-masing sektor besarnya 60° (mulai dari sektor satu sampai sektor enam). Pembagian keenam sektor dari bidang dq dapat dilihat pada Gambar 2.12. Sebuah vektor fluks stator dikatakan berhubungan dengan vektor tegangan ketika melewati sektor yang sama.



Gambar 2.12 Pembagian Ruang Vektor Tegangan Bidang dq

Pengaruh dari tiap-tiap vektor tegangan terhadap fluks stator dan torsi elektromagnetik dapat dilihat pada Tabel 2.1. Pengaruh dari vektor tegangan V_1 dan V_4 pada nilai torsi elektromagnetik adalah ambigu karena tergantung pada apakah vektor fluks mendahului atau tertinggal dari vektor tegangan. Vektor tegangan yang bernilai nol (V_0 dan V_7) tidak berpengaruh pada fluks stator akan tetapi menurunkan nilai torsi elektromagnetik.

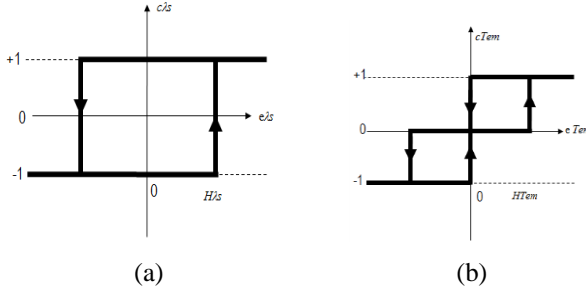
Tabel 2.1 Pengaruh Tiap-Tiap Vektor Tegangan terhadap Nilai Fluks Stator dan Torsi Elektromagnetik

	V_1	V_2	V_3	V_4	V_5	V_6	V_0 / V_7
λ_s	↗↗	↗	↘	↘↘	↘	↗	-
T_e	?	↗	↗	?	↘	↘	↘

2.3.1 Komparator Histerisis

Komparator digunakan untuk mendapatkan nilai kesalahan fluks dan torsi pada daerah histeresisnya. Untuk fluks digunakan *two level hysteresis* dan pada torsi digunakan *three level hysteresis*. Pada kesalahan fluks (*two-level hysteresis*), terdapat 2 nilai yaitu -1 atau 1 sedangkan pada kesalahan torsi (*three-level hysteresis*) dihasilkan 3 nilai yaitu -1, 0, 1. Nilai kesalahan (*error*) fluks dan torsi didapatkan dari operasi pengurangan antara nilai referensi dengan nilai estimasi.

Skema komparator histerisis fluks dan torsi elektromagnetik dapat dilihat pada Gambar 2.13.



Gambar 2.13 Kontroler Histerisis Fluks dan Kontroler Histerisis Torsi

2.3.2 *Estimator* dan Sektor Fluks Stator

Pada sistem DTC dibutuhkan umpan balik dari fluks dan torsi untuk dibandingkan dengan nilai referensinya. Umpan balik dari torsi bisa didapatkan tanpa harus menggunakan sensor dan sebagai gantinya adalah dengan mengestimasi nilainya dengan menggunakan estimator. Selain untuk mendapatkan nilai fluks dan torsi estimator digunakan untuk mendapatkan nilai sektor yang digunakan. Nilai magnitud dan sudut fluks stator estimasi dihitung dengan menggunakan Persamaan 2.25 dan Persamaan 2.26. Sedangkan untuk torsi estimasi didapatkan dari Persamaan 2.27

$$|\lambda_s| = \sqrt{\lambda_{ds}^2 + \lambda_{qs}^2} \quad (2.25)$$

$$\alpha = \angle \lambda_s = \tan^{-1} \frac{\lambda_{qs}}{\lambda_{ds}} \quad (2.26)$$

$$T_{em} = \frac{3P}{2} (\lambda_{as} \cdot i_{\beta s} - \lambda_{\beta s} \cdot i_{as}) \quad (2.27)$$

2.3.3 *Switching Table*

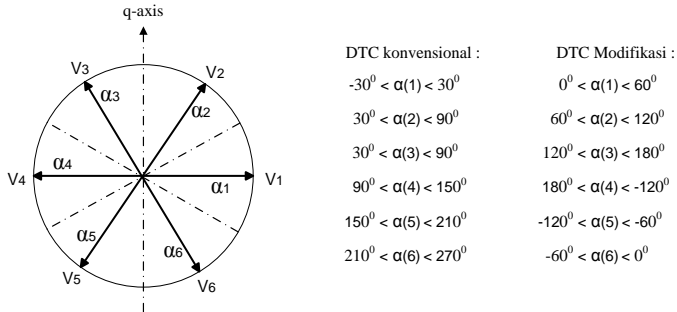
Switching Table digunakan untuk mendapatkn vektor tegangan yang digunakan yang pada nantinya menentukan *state inverter*. Masukan dari tabel *switching* adalah hasil keluaran dari komparator torsi dan komparator fluks. Nilai dari hasil keluaran komparator tersebut kemudian dibandingkan dengan sudut sektor yang didapatkan dari estimasi sektor. Dari ketiga nilai tersebut kemudian didapatkan nilai-nilai vektor tegangan yang digunakan berdasarkan tabel yang

telah disusun. Vektor tegangan terdiri dari vektor tegangan aktif dan vektor tegangan pasif. . Vektor tegangan aktif atau bernilai bukan nol terdiri dari $\bar{V}1$ (100); $\bar{V}2$ (110); $\bar{V}3$ (010); $\bar{V}4$ (011); $\bar{V}5$ (001); $\bar{V}6$ (101). Sedangkan vektor tegangan pasif atau bernilai nol adalah $\bar{V}0$ (000); $\bar{V}7$ (111). Aturan pemilihan tegangan pada tabel *switching* dapat dilihat pada Tabel 2.2

Tabel 2.2 *Switching Table* Vektor Tegangan Inverter

$H\psi$	HT_e	Sektor 1 $\alpha(1)$	Sektor 2 $\alpha(2)$	Sektor 3 $\alpha(3)$	Sektor 4 $\alpha(4)$	Sektor 5 $\alpha(5)$	Sektor 6 $\alpha(6)$
1	1	$\bar{V}2$	$\bar{V}3$	$\bar{V}4$	$\bar{V}5$	$\bar{V}6$	$\bar{V}1$
	0	$\bar{V}0$	$\bar{V}7$	$\bar{V}0$	$\bar{V}7$	$\bar{V}0$	$\bar{V}7$
	-1	$\bar{V}6$	$\bar{V}1$	$\bar{V}2$	$\bar{V}3$	$\bar{V}4$	$\bar{V}5$
-1	1	$\bar{V}3$	$\bar{V}4$	$\bar{V}5$	$\bar{V}6$	$\bar{V}1$	$\bar{V}2$
	0	$\bar{V}7$	$\bar{V}0$	$\bar{V}7$	$\bar{V}0$	$\bar{V}7$	$\bar{V}0$
	-1	$\bar{V}5$	$\bar{V}6$	$\bar{V}1$	$\bar{V}2$	$\bar{V}3$	$\bar{V}4$

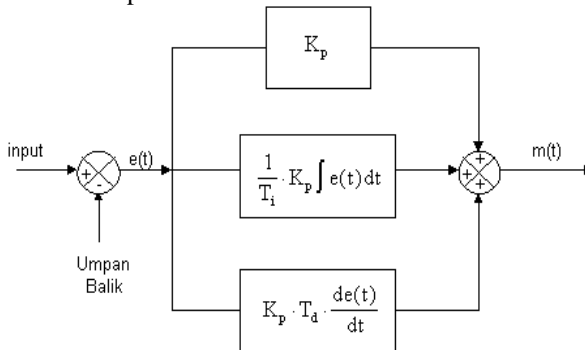
Switching Table pada Tabel. 2.2 digunakan pada DTC Konvensional. Pada DTC Konvensional *state inverter* pada V_1 dan V_4 tidak digunakan karena menurut Tabel 2.1 akan menghasilkan nilai torsi yang tidak jelas tergantung pada posisi sektor yang digunakan apakah berada pada posisi 30^0 pertama(yang menghasilkan torsi naik) atau pada posisi 30^0 kedua(yang menghasilkan torsi turun). DTC modifikasi adalah DTC konvensional yang dirubah atau digeser nilai vektornya. Pada DTC modifikasi *state inverter* yang tidak digunakan adalah V_3 dan V_6 karena nilai fluks stator pada kondisi *state* ini ambigu. Hal ini dianggap menjadi suatu keuntungan dalam mengontrol torsi dengan menggunakan metode DTC modifikasi. Lebih baik kehilangan penggunaan dua nilai fluks yang ambigu dibandingkan torsi. Perbedaan pembagian sektor antara DTC Konvensional dengan DTC Modifikasi dapat dilihat pada Gambar 2.14



Gambar 2.14 Sektor Tegangan *Inverter* dan Sektor *Switching* Fluks Stator pada DTC Konvensional dan DTC Modifikasi

2.4 Kontroler *Proporsional Integral Derivative*(PID)[6]

Kontroler PID terdiri dari Kontroler *Proporsional*, Kontroler *Integral*, dan Kontroler *Derivative*. Kontroler PID adalah kontroler dengan mekanisme umpan balik yang banyak digunakan pada bidang industri. Kontroler PID menghitung nilai kesalahan sebagai perbedaan antara variabel proses terukur dan *set point* yang diinginkan. Sampai saat ini Kontroler PID banyak digunakan karena tidak diperlukan teknik yang rumit dalam tahap perancangannya. Namun parameter yang ada pada kontroler PID harus diatur sesuai dengan sifat sistem yang digunakan. Jika sifat sistem berubah karena pengaruh dari hal-hal seperti usia penggunaan, perubahan beban yang signifikan dan sebagainya yang berpengaruh pada parameter sistem maka parameter dari kontroler PID perlu dirubah.



Gambar 2.15 Kontroler PID

Algoritma dari PID sendiri adalah dengan melibatkan 3 parameter yang terpisah yaitu nilai-nilai *Proporsional(P)*, *Integral(I)*, dan *Derivative(D)*. Pada ketiga parameter terdapat nilai K_p atau konstanta proporsional yang menjadi pengali bagi masing-masing P,I, dan D. Pada parameter P nilai *error* dikalikan Konstanta proporsional sehingga nilai P adalah proporsional dengan nilai *error* dari sistem.

Pada parameter I nilai *error* akan dikalikan dengan K_p dan dibagi dengan konstanta waktu T_i dan dikalikan dengan jumlah waktu pada saat kesalahan. Atau nilai tersebut di-*integral*-kan pada waktu. Nilai K_p dibagi dengan T_i sendiri dapat disebut dengan Konstanta *Integral* atau K_i . Kemudian pada parameter D nilai *error* dikurangi dengan nilai *error* pada waktu sebelumnya dan dikalikan dengan konstanta waktu T_d . Nilai K_p dikalikan dengan nilai T_d sendiri disebut dengan Konstanta *Derivative* atau K_d .

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2.28)$$

$$m(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int e(t) dt \quad (2.29)$$

dengan

$$K_i = \frac{K_p}{T_i}, \text{ dan } K_d = K_d \times T_d$$

Pengendali *proporsional* K_p akan memberikan efek mengurangi waktu naik tetapi tidak menghapus kesalahan keadaan tunak. Pengendali *integral* K_i akan memberikan efek menghapus kesalahan keadaan tunak tetapi berakibat memburuknya tanggapan *transient*. Pengendali *derivatif* K_d akan memberikan efek meningkatnya stabilitas sistem, mengurangi lewatan maksimum dan menaikkan tanggapan fungsi transfer. Efek dari masing-masing dapat dilihat pada Tabel 2.3.

Tabel 2.3 Perbandingan K_p , K_i , dan K_d

Respon Close-loop	<i>Rise Time</i>	<i>Overshoot</i>	<i>Setting Time</i>	<i>SS Error</i>
K_p	Menurun	Naik	Sedikit Peribahan	Menurun
K_i	Menurun	Naik	Naik	Tereliminasi

Respon Close- loop	<i>Rise Time</i>	<i>Overshoot</i>	<i>Setting Time</i>	<i>SS Error</i>
K_d	Sedikit Perubahan	Menurun	Menurun	Sedikit Perubahan

Dari Tabel 2.3 dapat diketahui bahwa pengendali *proporsional* akan mengurangi waktu naik, meningkatkan persentase lewatan maksimum dan mengurangi keadaan tunak. Sedangkan pengendali *proporsional derivatif* mereduksi lewatan maksimum dan waktu turun. Selain itu, pengendali *proporsional integral* menurun pada waktu naik, meningkatkan lewatan maksimum dan waktu turun dan akan menghilangkan kesalahan keadaan. Salah satu permasalahan terbesar dalam desain kontroler PID yaitu masalah *tunning* untuk menentukan nilai K_i , K_p , dan K_d yang pas. Metode-metode untuk melakukan *tunning* dilakukan berdasarkan model matematika sistem. Jika model tidak diketahui, maka dilakukan eksperimen terhadap sistem.

2.5 Sistem Pengaturan Cerdas[7]

Sistem pengaturan cerdas adalah salah satu cabang disiplin ilmu pengaturan dengan pendekatan yang unik. Pendekatan yang dilakukan lebih kepada basis proses pembelajaran dari pada basis pemodelan. Basis proses pembelejaraan adalah sekumpulan data yang akan menentukan bagaimana kontrol tersebut akan belajar nantinya.

Bagus atau tidaknya proses pembelajaran terlihat dari performa sistem yang meningkat. Respon yang baik ditunjukkan oleh konvergensi pembelajaran cepat, stabil, serta tetap kokoh terhadap berbagai perubahan parameter dan gangguan, baik internal dan eksternal. Konsep pengaturan cerdas sering digunakan pada banyak permasalahan seperti optimasi, stabilitas, regulator, dan *tracking*. Sehingga dalam konteks pengaturan sistem pengaturan cerdas mampu digunakan untuk sistem-sistem yang *nonlinear* yang pengontrolannya sulit dilakukan.

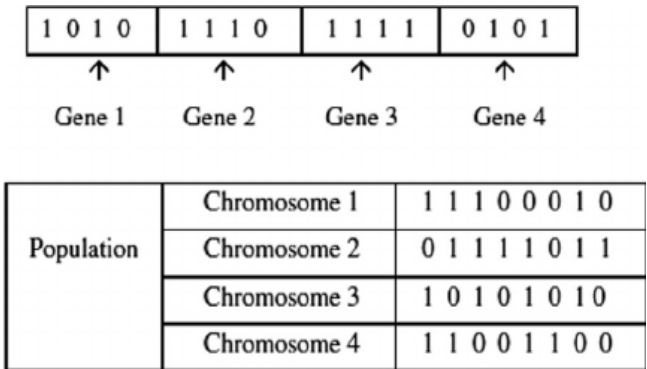
2.6 Genetic Algorithm(GA)[7],[8]

Genetic Algorithm merupakan suatu metode yang menggunakan seleksi alam yang merupakan bagian utama dari prinsip evolusi sebagai dasar pemikiran untuk menyelesaikan suatu permasalahan.

Prinsip ini berdasarka pemikiran yang dikemukakan oleh Charles Darwin, di mana tanpa menghiraukan prinsip dasar penurunan sifat, Darwin mengemukakan penggabungan kualitas induk pada generasi berikutnya, di samping itu bahwa individu yang mampu beradaptasi dengan lingkungannya akan mempunyai kesempatan hidup yang lebih besar.

Penggunaan prinsip genetika pada komputer dimulai pada tahun 1950 ketika beberapa ahli Biologi menggunakan komputer untuk simulasi sistem biologi. Akhir tahun 1975 John Holland dari Universitas Michigan melalui paper yang berjudul “*Adaption in Natural and Artificial System*” menggunakan konsep dasar Algoritma Genetika. Prinsip dari Algoritma Genetika adalah membangkitkan suatu kumpulan populasi yang berisi individu-individu dimana pada setiap individu-individu tersebut mempunyai suatu nilai ketahanan untuk bertahan hidup masing-masing.

Pada prosesnya *Genetic Algorithm* melakukan suatu proses kepada individu-individu tersebut dengan melihat nilai ketahanan individu tersebut yang akan menentukan individu yang terbaik. Masing-masing individu pada proses *Genetic Algorithm* mempunyai bagian yang menyusunnya yang disebut dengan kromosom dimana kromosom sendiri terdiri dari beberapa gen. Masing-masing gen mempunyai susunan alel yang berbeda yang menentukan sifat gen tersebut.



Gambar 2.16 Bagian individu pada *Genetic Algorithm*

Kromosom-kromosom tersebut pada nantinya akan mengalami perubahan seiring dengan proses GA yang akan menghasilkan individu-individu yang mempunyai sifat dan nilai ketahanan hidup yang berbeda dan menghasilkan generasi baru. Pada generasi baru masing-masing individu tadi mempunyai nilai ketahanan yang berbeda dan sifat yang berbeda. Iterasi untuk menghasilkan generasi baru akan terulang secara terus menerus sehingga didapatkan individu dengan parameter yang optimal sesuai yang diinginkan.

Nilai ketahanan pada proses *Genetic Algorithm* disebut dengan *fitness*. Nilai *fitness* adalah parameter optimal untuk menentukan individu terbaik yang akan dihasilkan. Parameter nilai *fitness* akan berbeda untuk setiap permasalahan optimalisasi yang dihadapi. Sehingga solusi yang menurut sebuah masalah adalah kondisi optimal, bukan berarti nilai tersebut adalah solusi optimal untuk permasalahan yang lain.

Dalam konteks pengaturan optimal Indeks Performansi adalah parameter yang sering digunakan untuk menentukan keoptimalan pengaturan sebuah sistem. Indeks Performansi banyak digunakan oleh kalangan akademisi untuk menentukan kualitas sebuah sistem. Indeks Performansi secara umum dapat dirumuskan dengan melihat kesalahan atau *error* pada sebuah sistem pengaturan yang dapat dituliskan pada Persamaan 2.30.

$$J = \int_0^{\infty} f(e)dt \quad (2.30)$$

Dimana J adalah Indeks Performansi dan e adalah nilai *error* atau kesalahan. Ada beberapa jenis Indeks Performansi yang menjadi parameter pada pengaturan performansi sebuah sistem diantaranya adalah *Integral of Absolute Error*(IAE). Pada IAE nilai *error* pada setiap waktu diambil nilai *absolute*-nya kemudian di-*integral*-kan. Rumus untuk IAE dapat dituliskan pada Persamaan 2.31.

$$J = \int_0^{\infty} |e(t)|dt \quad (2.31)$$

IAE sering digunakan karena secara matematis Indeks Performansi tersebut mudah dioperasikan. Dengan melihat nilai absolutnya maka dapat dengan mudah dilihat apakah nilai kesalahan yang terjadi besar atau kecil. Sistem optimal berdasarkan kriteria ini

biasanya mempunyai nilai redaman dan fasa *transient* yang sangat baik.

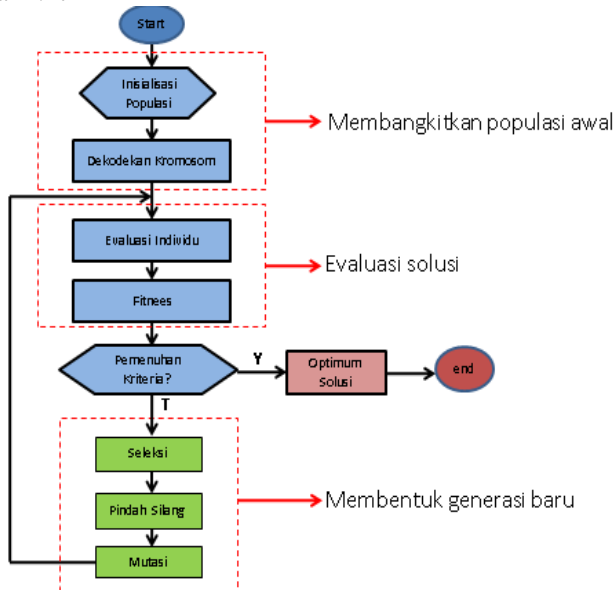
Pada proses *Genetic Algorithm* hal yang pertama dilakukan adalah membangkitkan populasi terlebih dahulu. Populasi terdiri dari puluhan atau ratusan individu yang masing-masing mempunyai keunikan dengan melihat dari kromosomnya. Representasi nilai kromosom yang sering digunakan dalam proses GA adalah bilangan biner. Bilangan biner sering dipakai karena akan memudahkan proses perubahan alel kromosom dikarenakan hanya terdapat dua kemungkinan nilai yaitu 1 dan 0.

Pembangkitan populasi dapat dilakukan dengan beberapa cara. Cara yang paling mudah adalah dengan membangkitkan populasi dengan nilai kromosom secara acak atau *random*. Dengan membangkitkan populasi secara acak maka terdapat banyak individu dengan kemungkinan nilai *fitness* yang banyak pula. Individu-individu tersebut kemudian masuk ke proses GA untuk diseleksi dan dilakukan pembentukan populasi yang baru pada generasi selanjutnya berdasarkan individu pada populasi pada generasi saat ini. Dalam pembentukan generasi baru individu-individu tersebut akan mengalami beberapa proses seleksi dan pengubahan nilai kromosom.

Sehingga dapat diibaratkan bahwa individu yang ada pada generasi sekarang merupakan orang tua pada generasi selanjutnya. Pada proses seleksi masing-masing individu dilihat nilai *fitness*-nya dan dicari nilai yang terbaik yang kemudian pada nantinya menghasilkan urutan individu berdasarkan nilai *fitness*-nya. Pada pembentukan individu baru masing masing individu generasi saat ini akan melakukan proses pertukaran alel gen sehingga pada masing-masing individu akan terjadi perubahan kromosom. Proses pertukaran alel gen ini disebut dengan *crossover*. Selanjutnya adalah proses GA melakukan perubahan pada setiap alel gen pada kromosom setiap individu yaitu dengan mengganti nilai pada alel gen tersebut. Dengan demikian akan terbangkit populasi yang mempunyai perbedaan karakteristik individu dengan generasi sebelumnya.

Namun demikian jika terjadi semua proses tersebut terdapat kemungkinan bahwa individu dengan nilai yang baik akan tidak terpilih atau terproduksi pada generasi selanjutnya. Sehingga pada kenyataannya adalah bisa dilakukan meminimalkan kemungkinan terjadinya proses perubahan pada alel. Semakin kecil nilai probabilitas pada proses pengubahan individu maka semakin kecil pula kemungkinan terjadinya perubahan individu pada pembentukan

generasi yang baru. Keseluruhan mekanisme GA ditunjukkan pada Gambar 2.17



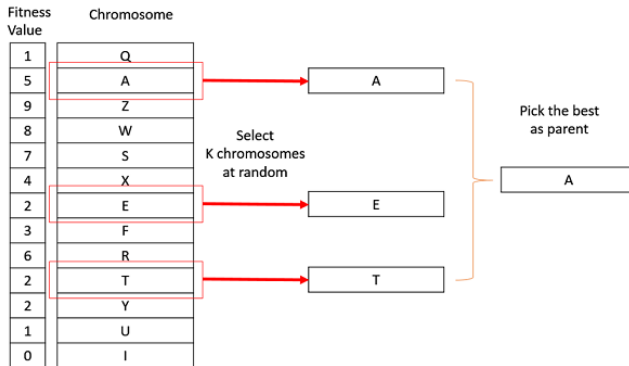
Gambar 2.17 Proses *Genetic Algorithm*

Dari Gambar 2.17 terlihat bahwa secara garis besar terdapat tiga tahap dalam proses *Genetic Algorithm* yaitu pembangkitan populasi awal, evaluasi individu untuk menentukan solusi, dan pembentukan generasi baru. Pada pembentukan generasi baru terdapat 3 proses yaitu seleksi, pindah silang atau *crossover*, dan mutasi. Individu baru yang terbentuk dari hasil pembentukan generasi baru disebut dengan *offspring*. Pada nantinya *offspring* yang telah dihasilkan akan digunakan untuk pembentukan generasi yang baru. Pada saat pembentukan generasi yang baru *offspring* akan berubah menjadi induk atau *parent*.

2.6.1 Seleksi

Proses seleksi pada GA adalah dengan memilih individu terbaik sesuai dengan nilai *fitness*-nya. Individu-individu terbaik dengan dilakukan proses seleksi pada nantinya akan mempunyai kemungkinan terpilih yang lebih besar. Terdapat beberapa macam

proses seleksi yang bisa dilakukan diantaranya adalah *roulette wheel*, *Tournament Selection*, *Steady-state reproduction*, *ranking and scalling*, dan *sharing*. Skema seleksi yang sering digunakan adalah *roulette wheel* yang dikemukakan oleh Holland. Kemudian seleksi yang juga cukup sering digunakan juga adalah *Tournament selection*.



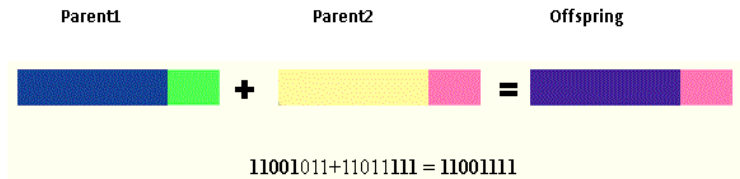
Gambar 2.18 Proses *Tournament Selection* GA

Tournament selection diperkenalkan oleh Goldberg, Korb, dan Deb. Metode ini dengan memilih secara acak beberapa kromosom dan membandingkan masing-masing nilai *fitness*-nya. Kromosom yang mempunyai nilai *fitness* yang terbaik akan dipilih dan dimasukkan pada urutan individu yang baru. Jumlah kromosom yang dipilih disebut dengan *tournament size*. Nilai *tournament size* yang sering digunakan adalah dua yang berarti bahwa dipilih dua kromosom dan membandingkannya. Pemilihan ini disebut dengan *binary tournament*. Dengan menggunakan seleksi *tournament* maka satu individu mempunyai kemungkinan untuk bisa terpilih lagi pada proses seleksi selanjutnya sehingga dimungkinkan satu individu tidak hanya dibandingkan oleh salah satu individu tetapi dapat juga dibandingkan dengan individu lainnya.

2.6.2 Crossover

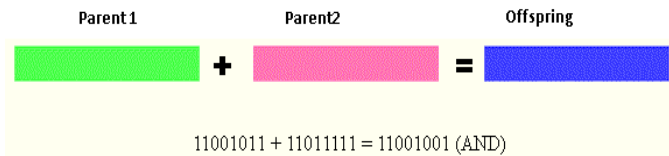
Proses *Crossover* pada *Genetic Algorithm* adalah dengan menukar salah satu alel pada gen kromosom dengan alel pada gen kromosom lainnya. Tipe-tipe *crossover* diantaranya adalah *single point crossover*, *two point crossover*, *uniform crossover*, dan *arithmetic crossover*. Penjelasan untuk *crossover* dicontohkan oleh

single point crossover. *Single point crossover* memilih alel kromosom antara awal kromosom sampai dengan titik *point crossover*. Alel yang dipilih kemudian saling bertukar dengan pasangan *crossover*-nya. *Single point crossover* ditunjukkan pada Gambar 2.19.



Gambar 2.19 *Single Point Crossover*

Mekanisme *crossover* yang paling sederhana adalah *arithmetic crossover*. *Arithmetic crossover* melakukan perpindahan silang berdasarkan operator aritmatika bahasa pemrograman seperti OR dan AND. Sehingga *arithmetic crossover* menghasilkan individu yang baru berbeda dengan kedua induknya. *Aritmetic Crossover* ditunjukkan pada Gambar 2.20.



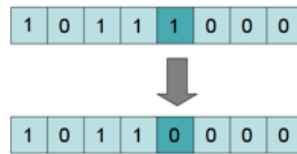
Gambar 2.20 *Arithmetic Crossover*

Namun demikian kemungkinan terjadinya *crossover* dapat dibatasi dengan menentukan sebuah nilai probabilitas yang disebut dengan probabilitas *crossover*. Kemudian dilakukan sebuah pembangkitan bilangan *random*. Jika bilangan *random* yang dibangkitkan kurang dari nilai probabilitasnya maka proses *crossover* dapat terjadi.

2.6.3 Mutasi

Proses mutasi mengubah alel-alel gen yang berada pada kromosom dengan alel-alel yang lain yang dipilih secara acak. Sehingga proses pada mutasi GA akan menghasilkan individu-individu yang mungkin tidak mungkin dihasilkan pada proses

crossover. Proses mutasi dilakukan untuk menjaga keragaman individu pada generasi yang baru. Proses mutasi ditunjukkan pada Gambar 2.21

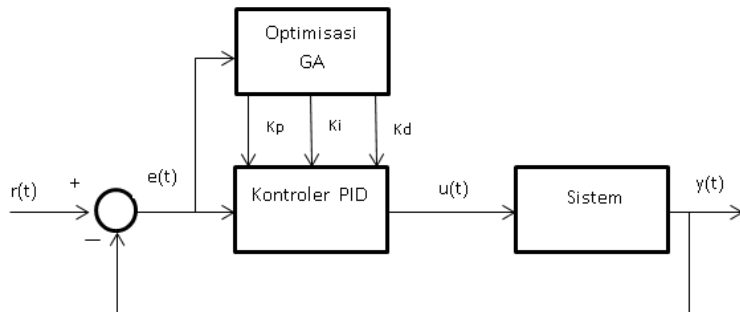


Gambar 2.21 Mutasi GA

Dengan representasi biner maka mutasi pada proses GA adalah dengan mengganti nilai bilangan biner pada titik mutasi yang terpilih. Jika pada titik tersebut alelnya bernilai 1 maka nilainya akan diganti dengan 0 begitu juga sebaliknya. Namun seperti pada proses *crossover* kemungkinan terjadinya mutasi dapat ditentukan dengan menentukan nilai probabilitas mutasinya. Jika nilai *random* yang dibangkitkan mempunyai nilai dibawah probabilitas mutasi maka mutasi dapat dilakukan jika nilainya berada diatas probabilitasnya maka mutasi tidak dapat dilakukan.

2.7 Kontroler PID-Genetic Algorithm(PID-GA)[9]

Kontroler PID-GA adalah kontroler dengan *tunning* parameter-parameternya dioptimalisasi dengan menggunakan *Genetic-Algorithm*. Dengan mekanisme *Genetic Algorithm* maka *tunning* dapat dilakukan dengan secara otomatis. Parameter untuk *tunning*-nya berdasarkan nilai keoptimalan dari fungsi objektifnya dalam hal ini adalah fungsi *fitness*-nya yang berdasarkan nilai *error*. Nilai *error* sendiri adalah perbedaan masukan yaitu *setpoint* dan keluaran yaitu hasil *output* sistem. Parameter yang di-*tunning* pada Kontroler PID-GA adalah nilai K_p , T_i , dan T_d dimana nilai awalnya ditentukan secara acak. Diagram dari kontroler PID-GA ditunjukkan pada Gambar 2.22.



Gambar 2.22 Kontroler PID-GA

Halaman ini sengaja dikosongkan

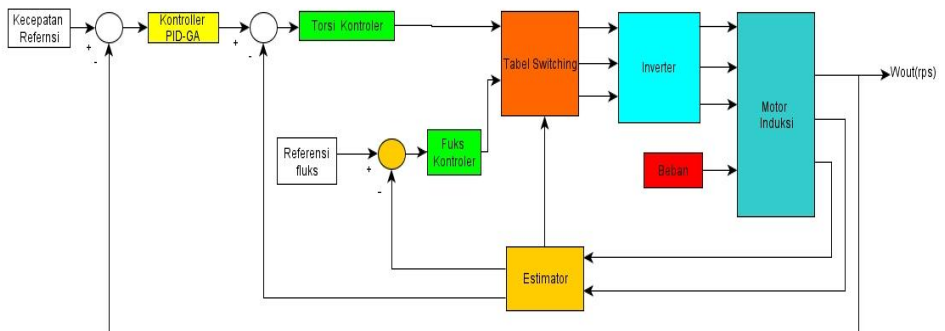
BAB 3

PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai tahapan-tahapan yang dilakukan dalam proses perancangan sistem secara detail. Proses perancangan yang dilakukan meliputi proses perancangan simulasi dari model matematika motor induksi, perancangan simulasi *Direct Torque Control* (DTC), perancangan simulasi pemodelan identifikasi motor induksi dengan DTC, dan yang terakhir yaitu perancangan simulasi kontroler PID berbasis Algoritma Genetika. Perancangan sepenuhnya dilakukan dengan menggunakan *Simulink* MATLAB.

3.1 Perancangan Blok Diagram

Blok diagram yang digunakan pada perancangan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Blok Simulasi

Pada Gambar 3.1 ditunjukkan diagram blok yang terdiri dari beberapa sub-sistem. Sistem diberikan kecepatan referensi yang menjadi pembanding untuk masukan kontroler. Blok Kontroler adalah Kontroler PID yang berbasis *Genetic Algorithm* yang digunakan untuk *men-tuning* nilai parameter-parameternya. Umpan balik dari kontroler adalah kecepatan keluaran dari motor induksi. Hasil dari kontroler PID-GA menjadi torsi referensi untuk dibandingkan dengan torsi estimasi yang didapatkan dari estimator. Hasil dari komparator kemudian dimasukkan ke torsi kontroler.

Estimator digunakan untuk mengestimasi torsi dan fluks yang digunakan. Masukan dari estimator didapatkan dari keluaran motor induksi yaitu hasil transformasi tegangan dan arus dari *inverter*. Hasil keluaran digunakan sebagai pembanding untuk kontroler torsi dan kontroler fluks yang nantinya akan dibandingkan dengan referensi masing-masing. Estimator juga mengeluarkan nilai sektor yang digunakan untuk menentukan vektor tegangan pada *switching table*.

Masukan dari *switching table* adalah hasil dari kontroler torsi, kontroler fluks, dan nilai sudut sektor dari estimator. Pada *switching table* sudah disusun sebuah tabel yang menentukan vektor tegangan yang digunakan. Vektor tegangan yang didapatkan kemudian digunakan untuk masukan *inverter*. Vektor tegangan yang dihasilkan dari kontroler digunakan untuk mengaktifkan saklar yang digunakan untuk *switching*.

Blok motor induksi berisi persamaan *state* pemodelan matematika dari motor induksi. Selain itu juga berisi transformasi persamaan 3 fasa ke persamaan 2 fasa. Output motor sendiri adalah arus, tegangan, dan kecepatan. Arus dan tegangan digunakan untuk masukkan estimator sedangkan kecepatan digunakan untuk masukan dari komparator untuk kontroler PID-GA.

3.2 Perancangan Model Motor Induksi 3 Fasa

Penurunan persamaan model matematika pada Subbab 2.2 yang telah dilakukan bisa menjadi referensi dalam perancangan model motor induksi yang akan dilakukan. Parameter-parameter dinamik diperlukan dalam pemodelan motor induksi. Parameter-parameter tersebut berdasarkan pada parameter-parameter fisik motor induksi yang akan digunakan. Pada tugas akhir ini, digunakan motor induksi tiga fasa Mohan dengan daya 3 HP/24KW. Tabel 3.1 menunjukkan parameter dari motor induksi tersebut.

Tabel 3.1 Parameter Motor Induksi 3 HP

No.	Nama	Nilai
1.	Tenaga Motor (HP)	3,0000
2.	Tegangan Motor (Line toLine) (Volt)	460,00

No.	Nama	Nilai
3.	Frekuensi (Hz)	60,000
4.	Jumlah pasang kutub	4,0000
5.	Tahanan Stator (Ohm)	1,7700
6.	Tahanan Rotor (Ohm)	1,3400
7.	Induktansi Stator (mH)	0,3826
8.	Induktansi Rotor (mH)	0,3808
9.	Induktansi Mutual (mH)	0,3687
10.	Momen Inersia (Kg.m ²)	0,0250

Dari parameter daya dan momen inersia pada motor pada Tabel 3.1 dengan kecepatan nominal yang digunakan adalah 120 rps maka dapat dicari beban nominal dari motor dengan Persamaan 3.1.

$$P = \frac{1}{2} J \omega^2 \quad (3.1)$$

$$P = \frac{1}{2} 0,025 \times 120^2$$

$$P = 2400 - 180 = 2220 \text{ Watt}$$

Daya 2220 Watt merupakan daya keluaran motor yang dapat diberikan pada motor. Daya tersebut adalah daya keluaran yang dapat memenuhi beban yang diberikan pada motor. Sehingga bisa didapatkan beban nominal dengan menggunakan Persamaan 3.2.

$$P = T_{nom} \times \omega \quad (3.2)$$

$$T_{nom} = \frac{P}{\omega} \quad (3.3)$$

$$T_{nom} = \frac{2220}{120} = 18,5 \text{ Nm}$$

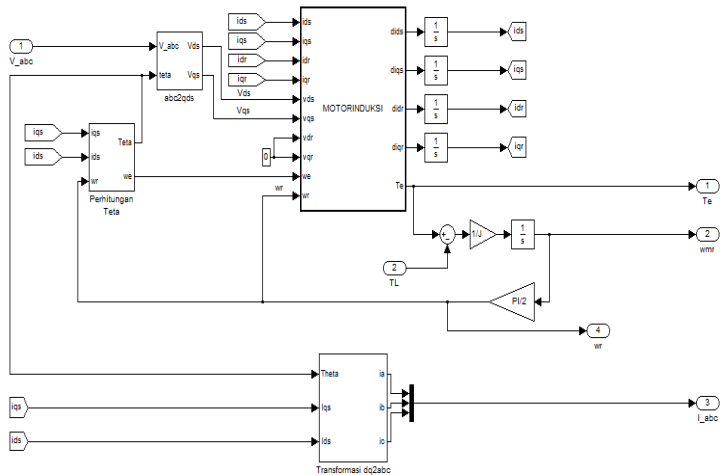
Torsi diatas merupakan torsi nominal dari motor tersebut sehingga dapat diperoleh torsi maksimal yaitu 20-25% dari torsi nominal.

$$T_{max} = T_{nom} + \frac{20}{100} T_{nom} = 18,5 + \frac{20}{100} 18,5 = 22,2 \text{ Nm}$$

Model matematis dari motor induksi yang digunakan dalam perancangan ini didapatkan dari Persamaan 2.16 yaitu:

$$\frac{d}{dt} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} = \mathbf{Q}^{-1} \begin{bmatrix} V_{qs} \\ V_{ds} \\ V_{qr} \\ V_{dr} \end{bmatrix} - \mathbf{Q}^{-1} \mathbf{P} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix}$$

Dari persamaan *state* arus tersebut maka bisa digunakan untuk mencari nilai arus dari motor induksi. Persamaan *state* arus tersebut kemudian dimasukkan ke dalam model motor induksi pada *Simulink*. Parameter-parameter pada Tabel 3.1 juga pada nantinya dimasukkan ke model *Simulink* pada MATLAB. Model motor induksi 3 fasa pada *Simulink* dapat dilihat pada Gambar 3.2.



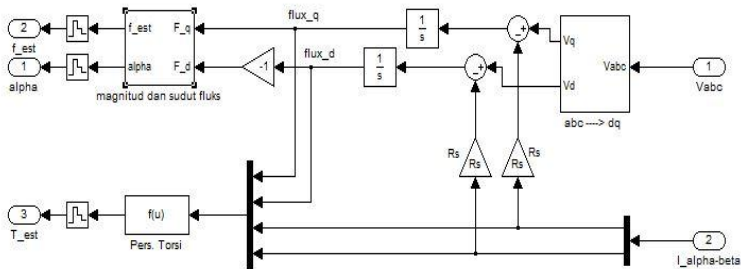
Gambar 3.2 Model *Simulink* Motor Induksi

3.3 Perancangan *Direct Torque Control*(DTC) Motor Induksi

Selanjutnya adalah melakukan perancangan DTC untuk kontrol motor induksi. DTC terdiri dari beberapa sub bagian yang masing-masing bagian mempunyai pemodelannya sendiri. Input dari DTC sendiri adalah referensi torsi dan referensi fluks. DTC mempunyai fungsi untuk mempertahankan nilai torsi dan nilai fluks tetap pada referensinya. Nilai referensi torsi sendiri adalah hasil sinyal kontrol dari kontroler yang digunakan. Sedangkan nilai referensi fluks berupa nilai konstan dengan besar 0,96. Keluaran dari DTC yang berupa vektor tegangan kemudian terhubung dengan *inverter* 3 fasa yang pada nantinya terhubung dengan motor induksi.

3.3.1 Perancangan *Estimator* dan Sektor Sudut Fluks

Kelahiran dari DTC adalah bisa didapatkan nilai kecepatan tanpa harus menggunakan sensor kecepatan sebagai gantinya adalah digunakan *estimator*. *Estimator* pada DTC digunakan untuk mendapatkan nilai umpan balik torsi dan fluks dengan berdasar pada nilai tegangan dan nilai arus dari *inverter*. Model *Simulink* untuk *estimator* dapat dilihat pada Gambar 3.3.

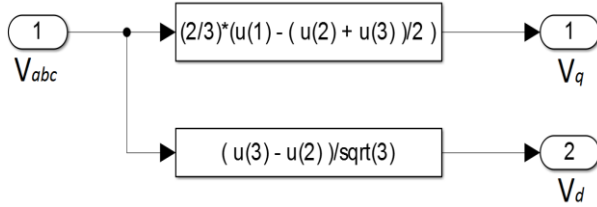


Gambar 3.3 Perancangan *Estimator* DTC

Nilai estimasi fluks pada nantinya dibandingkan dengan fluks referensi pada fluks kontroler dan nilai estimasi torsi pada nantinya dibandingkan dengan nilai torsi referensi pada torsi kontroler. Nilai sudut sektor yang didapat diberikan ke tabel *switching* sehingga dapat dibandingkan dengan nilai torsi dan nilai fluks. Persamaan yang

digunakan untuk mendapatkan torsi estimasi didapat dari Persamaan 2.27 dan untuk mendapatkan fluks referensi didapatkan dari Persamaan 2.25. untuk mendapatkan nilai sudut sektor digunakan Persamaan 2.26.

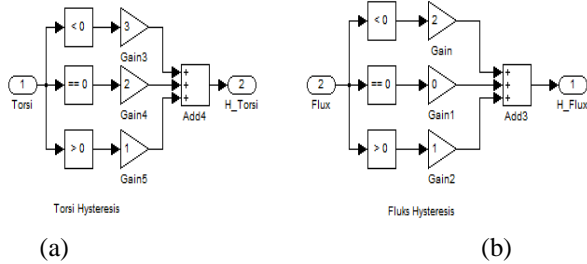
Persamaan 2.20 digunakan untuk membuat blok ‘abc→dq’ pada blok *estimator* dan dibuat dalam fungsi MATLAB untuk perancangan simulasinya seperti pada Gambar 3.4.



Gambar 3.4 Transformasi Tegangan dari abc ke dq

Perancangan sektor sudut fluks stator bertujuan untuk mengubah besarnya sudut fluks stator(α) menjadi dalam bentuk sektor. Sektor sudut fluks stator dibagi menjadi enam dan masing-masing sektor besarnya 60^0 . Besarnya masing-masing sektor dapat dilihat pada Tabel 2.2 dapat dibuat rancangan simulasi sektor sudut fluks stator yang dapat dilihat pada Gambar 3.5.

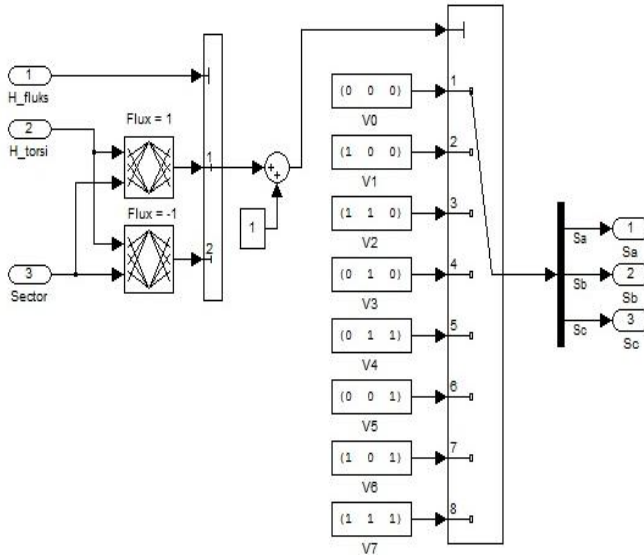
Pada Gambar 3.5 dapat dilihat bahwa sinyal masukan berupa besarnya sudut fluks stator yang dihasilkan estimator akan ditentukan letaknya sesuai sektornya. Nilai sudut fluks stator akan dibandingkan nilainya dengan besarnya sudut untuk masing-masing sektor. Sektor sudut fluks stator ditentukan dari posisi nilai sudut fluks stator itu berada sesuai dengan batas untuk masing-masing sektor pada Tabel 2.2. Sinyal keluaran berupa nilai satu sampai enam yang masing-masing mewakili nilai keenam sektor.



Gambar 3.6 Blok Komparator Torsi dan Blok Komparator Fluks

3.3.3 Perancangan Blok *Switching Table*

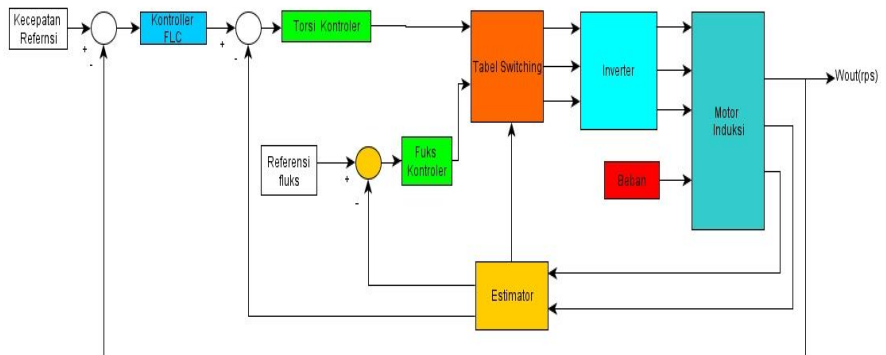
Blok *switching table* digunakan untuk memilih vektor tegangan yang aktif berdasarkan kondisi yang diperoleh dari blok komparator fluks dan torsi serta sektor dari blok sektor sudut fluks stator. Kondisi-kondisi tersebut kemudian disesuaikan dengan Tabel 2.1 sehingga nantinya terdapat 8 vektor yang akan menentukan kondisi *switch* dari *inverter*. Gambar 3.7 menunjukkan blok *switching table* pada simulasi *Direct Torque Control*.



Gambar 3.7 Blok *Switching Table*

3.4 Perancangan Kontroler *Fuzzy Logic Controller*(FLC)-DTC[10],[11]

Perancangan pengaturan motor induksi dengan menggunakan *Fuzzy Logic Controller*(FLC) digunakan sebagai pembanding untuk perancangan Kontroler PID-GA yang akan dilakukan. Blok diagram untuk perancangan FLC-DTC ditunjukkan pada Gambar 3.8

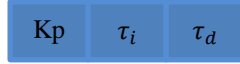


Gambar 3.8 Blok Diagram *Fuzzy Logic Controller* untuk DTC

Simulasi yang dilakukan merupakan gabungan dari perancangan model motor induksi ditambah dengan *Direct Torque Control*(DTC) pada model motor induksi tersebut ditambahkan *Fuzzy Logic Controller*(FLC) untuk mengatur masukan yang berupa nilai referensi kecepatan yang akan diubah menjadi torsi dan menjadi masukan untuk blok kontroler torsi histerisis yang dapat memberikan respon sesuai dengan nilai acuan/referensi yang diberikan pada Kontroler FLC.

3.5 Perancangan Kontroler PID-GA[9]

Kontroler yang digunakan pada pengerjaan tugas akhir ini adalah dengan menggunakan PID dengan parameternya di-tuning dengan menggunakan Algoritma Genetika untuk mendapatkan kontroler yang optimal. Ukuran kromosom yang digunakan adalah 30 bit dengan masing-masing parameter bernilai 10 bit. Sehingga pada satu individu calon optimal terdapat tiga nilai yaitu K_p , τ_i , dan τ_d . Kromosom pada suatu calon individu dapat digambarkan seperti pada Gambar 3.9



Gambar 3.9 Representasi Kromosom Individu

Tipe seleksi yang digunakan pada perancangan GA ini adalah *tournament*, dengan *crossover* menggunakan tipe *singlepoint*, dan mutasi *bitflip*. Fungsi objektif yang digunakan pada perancangan Kontroler PID–GA ini adalah dengan menggunakan IAE. Indeks performansi dituliskan dengan menggunakan Persamaan 3.4

Nilai *fitness* didapatkan dengan Persamaan 3.4

$$f = \frac{1}{1+J} \quad (3.4)$$

Individu-individu yang dihasilkan menggunakan *Genetic Algorithm* merupakan deretan bilangan biner. Sehingga agar dapat diterapkan pada kontroler deretan bilangan biner tersebut dirubah ke bilangan desimal. Pada prosesnya terdapat *range* nilai maksimum dan minimum yang dibolehkan untuk digunakan pada masing-masing parameter sehingga perlu dirancang suatu skema rumus pengubahan bilangan. Rumus yang digunakan untuk merubah bilangan biner ke desimal ditunjukkan pada Persamaan 3.5.

$$x_i = x_{min} + \frac{x_{max} - x_{min}}{2^L - 1} y_i (i = 1, 2, 3) \quad (3.5)$$

x_{min} adalah nilai minimal dari *range* dan x_{max} adalah nilai maksimum dari parameter. L adalah nilai bit yang digunakan dan y_i adalah variabel faktor pengali.

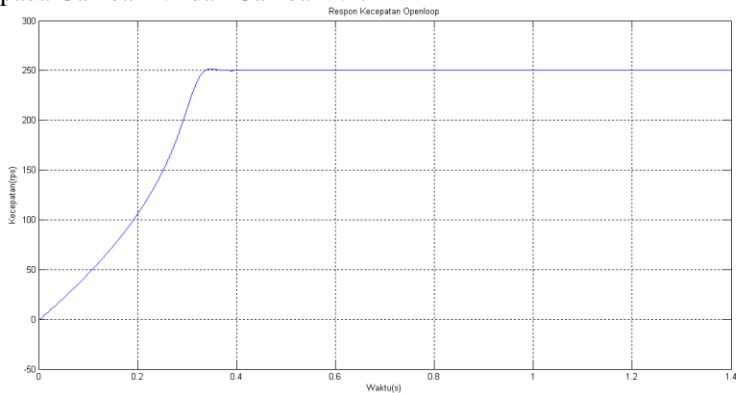
BAB 4

SIMULASI DAN HASIL

Pada bab ini akan dibahas mengenai simulasi dari hasil perancangan sistem pada BAB 3. Setelah dilakukan simulasi kemudian dilakukan analisis terhadap data hasil simulasi apakah sesuai dengan target dari tujuan penelitian.

4.1 Simulasi Model Motor Induksi

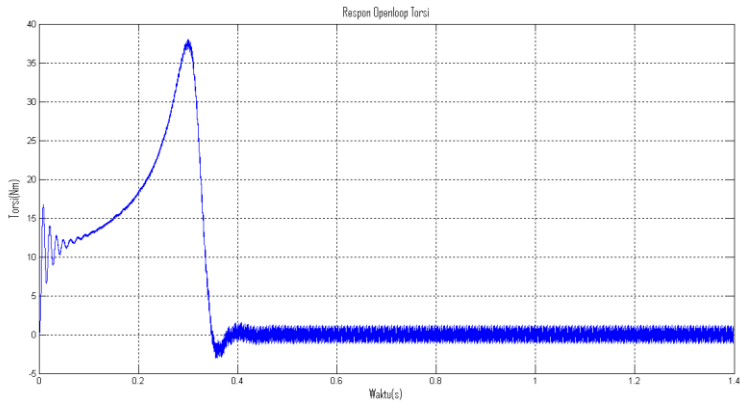
Tahapan simulasi ini dilakukan untuk mengetahui respon torsi elektromagnetik dan kecepatan rotor dari hasil perancangan model motor induksi pada pembahasan perancangan sistem Subbab 3.1. Respon dari torsi elektromagnetik dan kecepatan rotor dapat dilihat pada Gambar 4.1 dan Gambar 4.2.



Gambar 4.1 Respon Kecepatan Rotor

Dari respon didapatkan spesifikasi sebagai berikut :

Kecepatan nominal	= 250,5 rps
<i>Time constan</i> (τ)	= 0,2157 detik
Waktu mencapai <i>steady state</i>	= 0,321 detik
<i>Overshoot</i> (M_p)	= 0,16%
<i>Rise time</i> (t_r (5%-95%))	= 0,2923 detik



Gambar 4.2 Respon Torsi Rotor

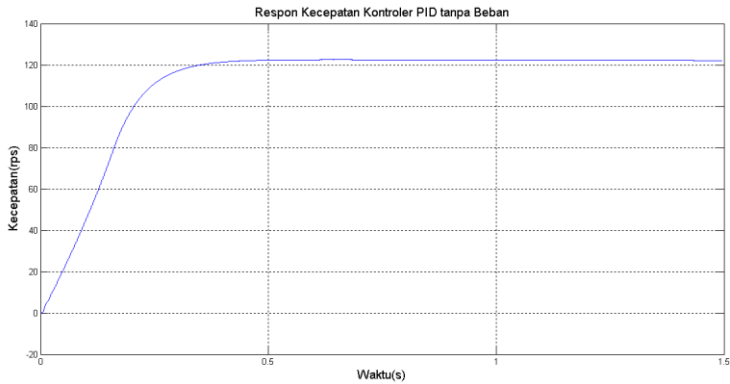
Pada respon torsi menunjukkan waktu *starting* motor sebesar 0,3 detik dengan nilai torsi awal sebesar 37 Nm. Nilai torsi elektromagnetik awal yang besar dikarenakan nilai arus stator *starting* motor yang dibangkitkan sangat besar. Namun *ripple* tersebut semakin mengecil hingga bernilai sekitar nol dan mencapai *steady state* ketika $t = 0,4$ detik.

4.2 Simulasi dan Pengujian *Direct Torque Control* (DTC) dengan Kontroler PID

Pada dasarnya metode *Direct Torque Control*(DTC) menggunakan kontroler PID sebagai kontroler kecepatan. Pada subbab ini akan mensimulasikan metode *Direct Torque Control* dengan kontrol PID pada *plant* motor induksi yang telah dirancang. Nilai parameter kontroler PID diperoleh dari hasil *tunning* dengan nilai $K_p=1,44$, $K_i=0,29$, $K_d=0,05$.

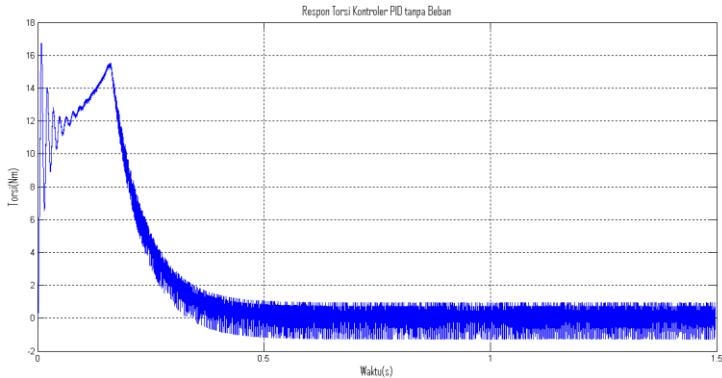
4.2.1 Simulasi dan Pengujian Tanpa Beban

Simulasi dilakukan pada kondisi tanpa beban untuk melihat karakteristik kontrol PID pada pengaturan kecepatan motor induksi tiga fasa. Gambar 4.3 menunjukkan respon kecepatan motor induksi dengan kontroler PID pada kondisi tanpa beban.



Gambar 4.3 Respon Kecepatan Kontroler PID Tanpa Beban

Dari Gambar 4.3 dapat diketahui bahwa motor mencapai *steady state* pada 0,54 detik dengan *rise time* 0,47 detik dengan kecepatan motor *steady state* yaitu 119,8 rps. Sehingga *error steady state* sebesar 0,167 %. Kemudian dilihat respon dari motor induksi dengan menggunakan Kontroler PID. Hasil respon torsi ditunjukkan pada Gambar 4.4

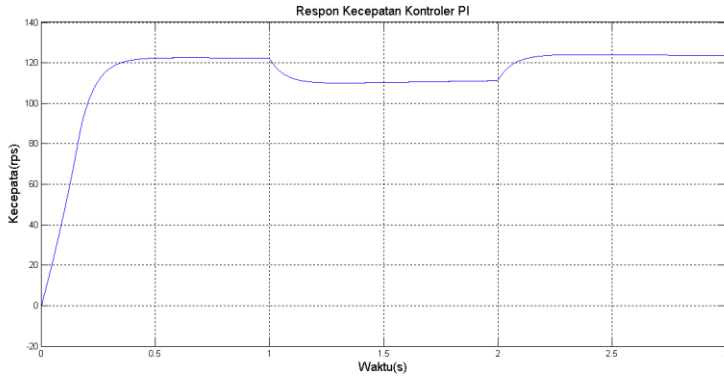


Gambar 4.4 Respon Torsi Kontroler PID Tanpa Beban

Dari Gambar 4.4 torsi yang dihasilkan motor mengikuti torsi referensi yang diberikan dengan *ripple* pada torsi aktual sebesar $\pm 2,17 \text{ Nm}$.

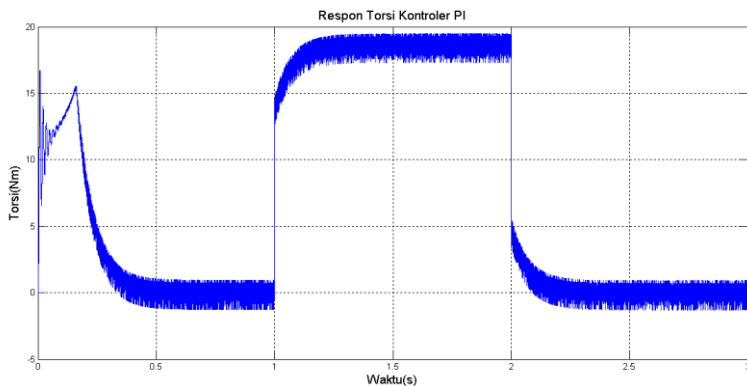
4.2.2 Pengujian dengan Menggunakan Perubahan Beban

Simulasi dengan pengujian beban dilakukan dengan memberikan beban langsung pada motor. Pengujian dilakukan dengan memberikan beban nominal. Nilai nominal telah dihitung dan ditentukan pada BAB 3



Gambar 4.5 Respon Kontroler PID Terbeban

Beban diberikan pada saat $t=1$ detik kemudian dilepas pada saat $t=2$ detik. Dari Gambar 4.5 dapat dilihat terjadi penurunan kecepatan sebesar 10 rps pada kecepatan disaat *steady state*. Untuk respon torsi ditunjukkan seperti pada Gambar 4.6.

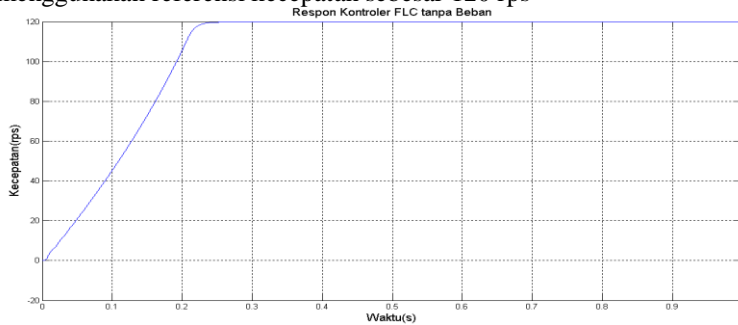


Gambar 4.6 Respon Torsi Kontroler PID Terbeban

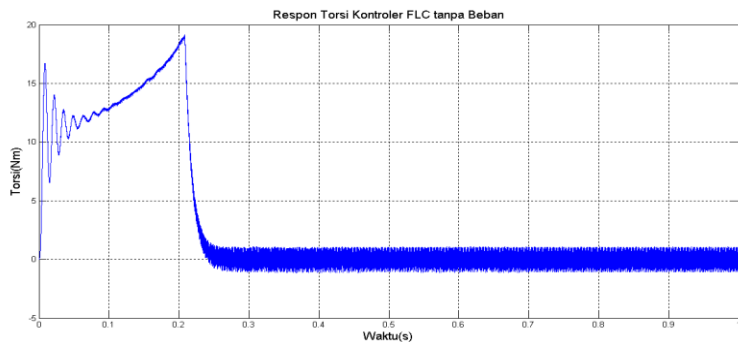
Pengujian beban dilakukan dengan memberikan beban dengan Nilai torsi beban nominal pada saat $t=1$ detik dan dilepas pada saat $t=2$ detik. Dari Gambar 4.6 dapat dilihat bahwa respon torsi mengikuti nilai torsi beban yang diberikan dengan waktu mencapai keadaan *steady state* setelah beban diberikan sebesar 0,32 detik

4.3 Simulasi dan Pengujian *Direct Torque Control* (DTC) dengan *Fuzzy Logic Controller*(FLC)

Pengujian untuk dengan menggunakan. *Fuzzy Logic Controller* dilakukan sama seperti dengan menggunakan Kontroler PID yaitu dengan melihat respon kecepatan dan respon torsi saat tidak diberi beban dan diberi beban. Respon kecepatan motor saat tidak diberi beban ditunjukkan pada Gambar 4.7 dan respon torsi saat diberi beban ditunjukkan pada Gambar 4.8. Pengujian dilakukan dengan menggunakan referensi kecepatan sebesar 120 rps

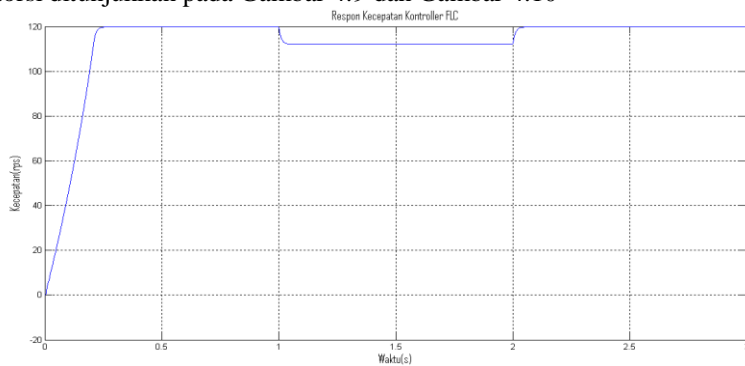


Gambar 4.7 Respon Kecepatan Kontroler FLC tanpa Beban

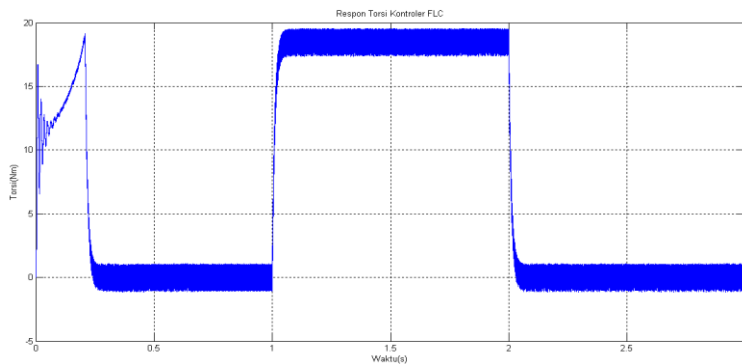


Gambar 4.8 Respon Torsi Kontroler FLC tanpa Beban

Dari Gambar 4.7 dapat dilihat bahwa didapatkan respon kecepatan yang lebih cepat mencapai *steady state* jika dibandingkan dengan respon kontroler PID dengan waktu mencapai *steady state* sebesar 0,21 detik. Dari Gambar 4.8 juga dapat dilihat bahwa jika dibandingkan dengan menggunakan kontroler PID didapatkan respon torsi yang lebih cepat mencapai *steady state* namun didapatkan lonjakan torsi awal yang lebih besar sebesar 18,5 Nm. Kemudian dilakukan pengujian dengan menggunakan beban. Beban yang digunakan adalah beban nominal. Beban diberikan pada saat $t = 1$ detik kemudian dilepas pada saat $t = 2$ detik. Respon kecepatan dan respon torsi ditunjukkan pada Gambar 4.9 dan Gambar 4.10



Gambar 4. 9 Respon Kecepatan Kontroler FLC Terbeban



Gambar 4.10 Respon Torsi Kontroler FLC Terbeban

Pada Gambar 4.9 dapat dilihat bahwa saat diberikan beban respon kecepatan turun sebanyak 10 rps. Nilai penurunan ini sama dengan respon kontroler PID namun mencapai nilai *steady* yang lebih cepat. Dari Gambar 4.10 juga diperlihatkan bahwa respon torsi kontroler FLC lebih cepat mencapai *steady state* dengan Nilai *ripple* yang hampir sama dengan *ripple* torsi hasil kontroler PID.

4.4 Simulasi dan Pengujian *Direct Torque Control* (DTC) dengan Kontroler PID-GA

Pada subbab ini, akan disimulasikan metode *Direct Torque Control*(DTC) dengan kontroler PID berbasis *Genetic Algorithm*. Pada pengujiannya, rancangan akan disimulasikan baik tanpa perubahan beban dan dengan perubahan beban.

4.4.1 Proses *Learning* Algoritma Genetika

Penalaan Parameter PID dengan menggunakan Algoritma Genetika dilakukan secara *offline* sehingga perlu dilakukan identifikasi pemodelan terlebih dahulu. *Plant* yang diidentifikasi adalah Blok DTC dan motor induksi. Identifikasi dilakukan dengan melihat nilai respon *open loop* dari blok DTC dan motor Induksi.

Hasil identifikasi yang diperoleh kemudian dicari nilai MSE-nya. Hasil dari identifikasi ditunjukkan pada Tabel 4.1.

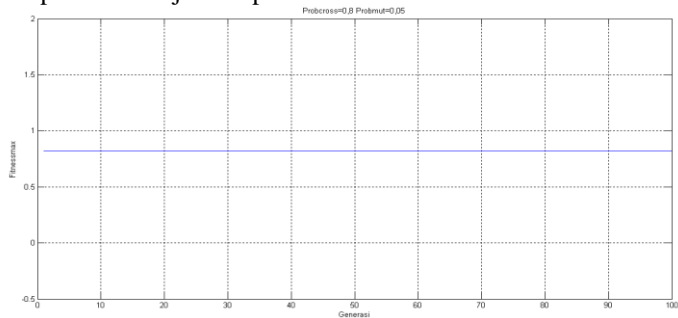
Tabel 4.1 Hasil Identifikasi sistem

No	Hasil Identifikasi	MSE
1	$G(s) = \frac{2,08}{0,0149s^2 + 0,2457s + 1}$	1,6081
2	$G(s) = \frac{2,08}{6,81 \times 10^{-4}s^3 + 0,022s^2 + 0,258s + 1}$	1,5615
3	$G(s) = \frac{2,08}{0,03354s^3 + 0,312s^2 + 0,9675s + 1}$	6,7345
4	$G(s) = \frac{2,08}{5,4 \times 10^{-5}s^3 + 0,02025s^2 + 0,2465s + 1}$	1,3486
5	$G(s) = \frac{2,08}{3,19 \times 10^{-3}s^2 + 0,113s + 1}$	1,5433

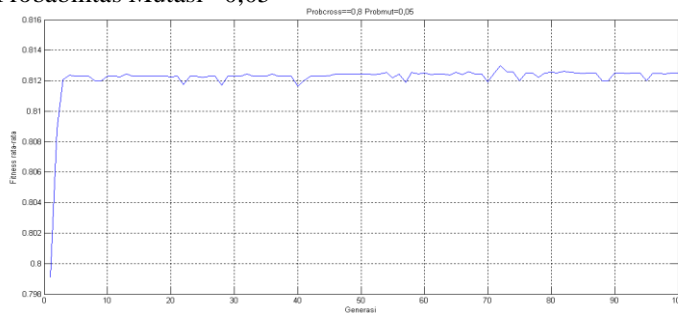
Dari Tabel 4.1 dapat dilihat hasil identifikasi yang mempunyai nilai MSE yang paling kecil sebesar 1,3486. Model matematika ini yang pada nantinya digunakan untuk proses *learning* sebagai representasi dari Blok DTC dan Motor Induksi.

Hasil Pemodelan yang sudah didapatkan digunakan pada proses *learning* oleh Algoritma Genetika sehingga didapatkan parameter yang optimal. Pada proses *learning* dilakukan beberapa kali percobaan dengan mengubah parameter parameter yang ada seperti nilai Probabilitas *Crossover*, Probabilitas Mutasi, dan Generasi. Nilai Parameter maksimal untuk $K_p=50$, $K_i=0,15$, $K_d=4$ dan nilai minimalnya adalah masing-masing $K_p=30$, $K_i=0,1$ dan $K_d=0,01$.

Percobaan pertama dilakukan dengan menggunakan nilai Probabilitas *Crossover* 0,8 dan Probabilitas Mutasi 0,05. Nilai *fitness* dilihat untuk melihat konvergensi nilainya. Grafik *fitness* yang didapatkan ditunjukkan pada Gambar 4.11

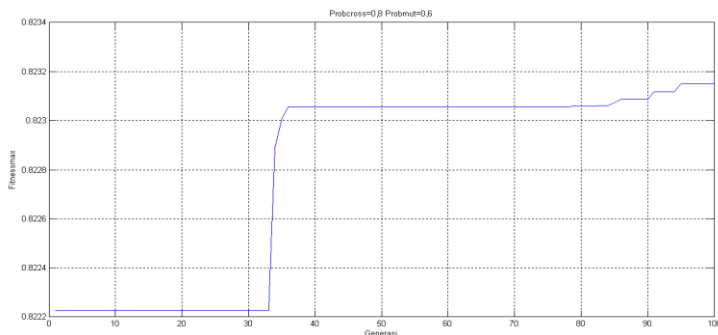


Gambar 4.11 Nilai *Fitness* Maksimal Probabilitas *Crossover*=0,8 Probabilitas Mutasi =0,05

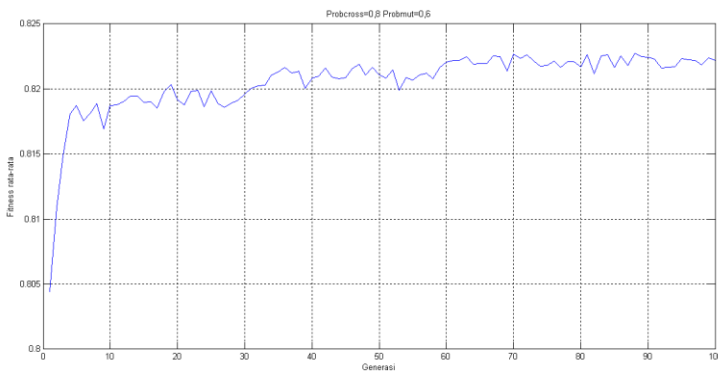


Gambar 4.12 Nilai *Fitness* Rata-rata Probabilitas *Crossover*=0,8 Probabilitas Mutasi =0,05

Hasil *tunning* yang didapatkan adalah $K_p=45,91$, $K_i=0,1023$, dan $K_d= 1,4297$. Dengan nilai *fitness* maksimalnya adalah 0,8216. Kemudian dilakukan dengan menggunakan nilai Probabilitas *Crossover* 0,8 dan Probabilitas Mutasi 0,6. Grafik *fitness* yang didapatkan ditunjukkan pada Gambar 4.13



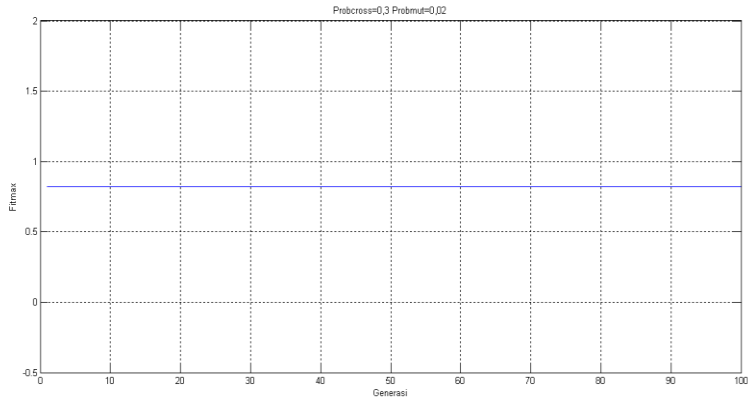
Gambar 4.13 Nilai *Fitness* Maksimal Probabilitas *Crossover*=0,8 dan Probabilitas Mutasi =0,6



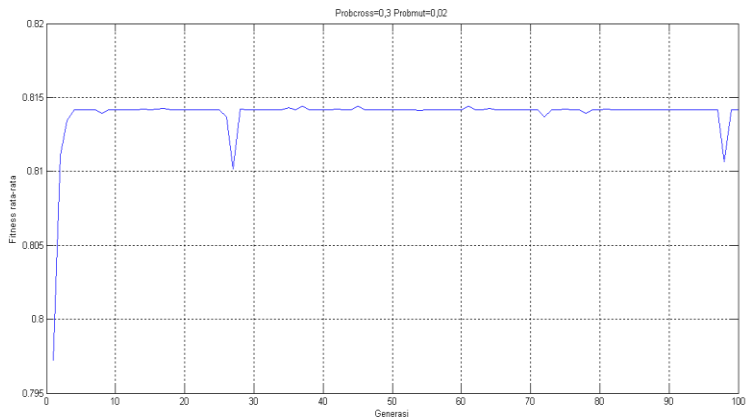
Gambar 4.14 Nilai *Fitness* Rata-rata Probabilitas *Crossover*=0,8 dan Probabilitas Mutasi =0,6

Hasil *tunning* yang didapatkan adalah $K_p=50$, $K_i=0,1202$, dan $K_d=3,95$. Dengan nilai *fitness* maksimalnya adalah 0,8231. Kemudian dilakukan dengan menggunakan nilai Probabilitas *Crossover* 0,3 dan

Probabilitas Mutasi 0,02. Grafik *fitness* yang didapatkan ditunjukkan pada Gambar 4.15



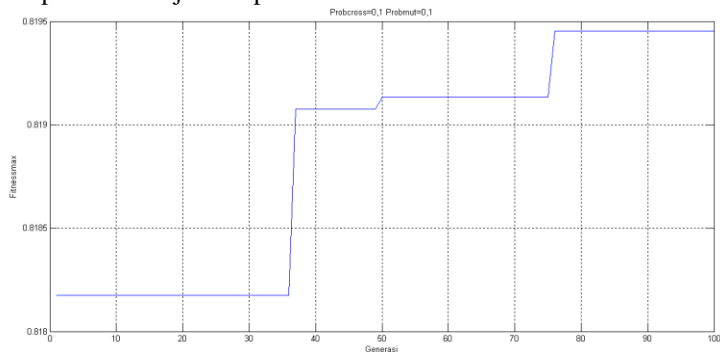
Gambar 4.15 Nilai *Fitness* Maksimal Probabilitas *Crossover*=0,3 dan Probabilitas Mutasi =0,02



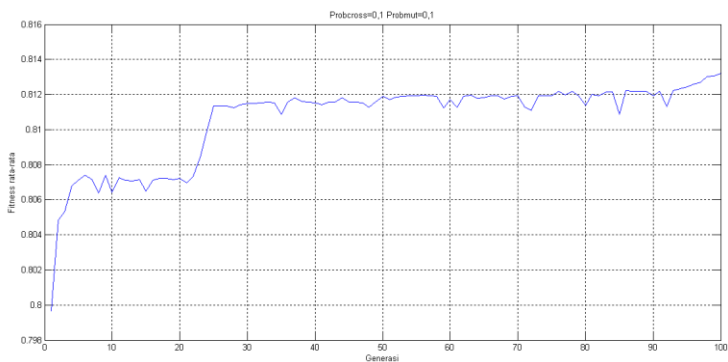
Gambar 4.16 Nilai *Fitness* Rata-rata Probabilitas *Crossover*=0,3 dan Probabilitas Mutasi =0,02

Hasil *tunning* yang didapatkan adalah $K_p=48,9052$, $K_i=0,1242$, dan $K_d=3,9922$. Dengan nilai *fitness* maksimalnya adalah 0,8231. Kemudian dilakukan dengan menggunakan nilai Probabilitas

Crossover 0,1 dan Probabilitas Mutasi 0,1. Grafik *fitness* yang didapatkan ditunjukkan pada Gambar 4.17

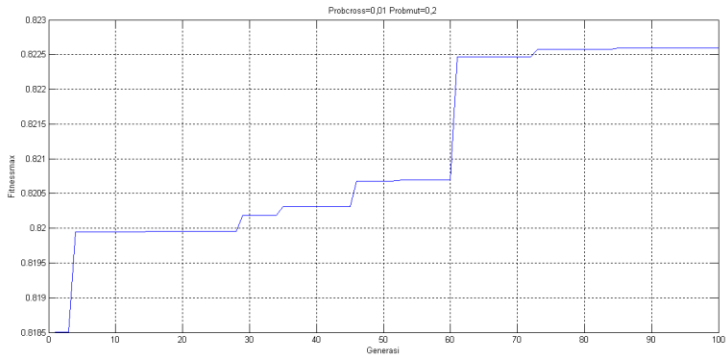


Gambar 4.17 Nilai *Fitness* Maksimal Probabilitas *Crossover*=0,1 dan Probabilitas Mutasi =0,1

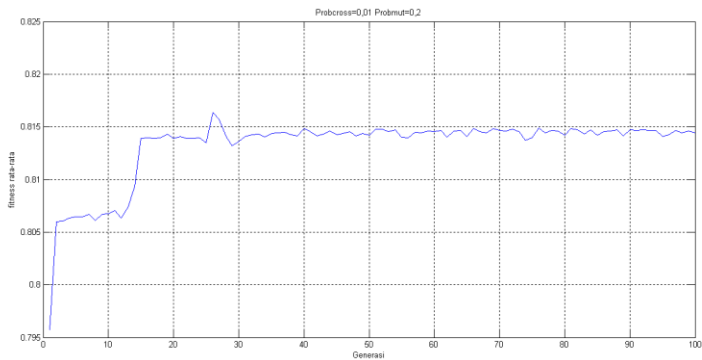


Gambar 4.18 Nilai *Fitness* Rata-rata Probabilitas *Crossover*=0,1 dan Probabilitas Mutasi =0,1

Hasil *tunning* yang didapatkan adalah $K_p=43,3138$, $K_i=0,1242$, dan $K_d=3,9922$. Dengan nilai *fitness* maksimalnya adalah 0,8195. Kemudian dilakukan dengan menggunakan nilai Probabilitas *Crossover* 0,01 dan Probabilitas Mutasi 0,2. Grafik *fitness* yang didapatkan ditunjukkan pada Gambar 4.19

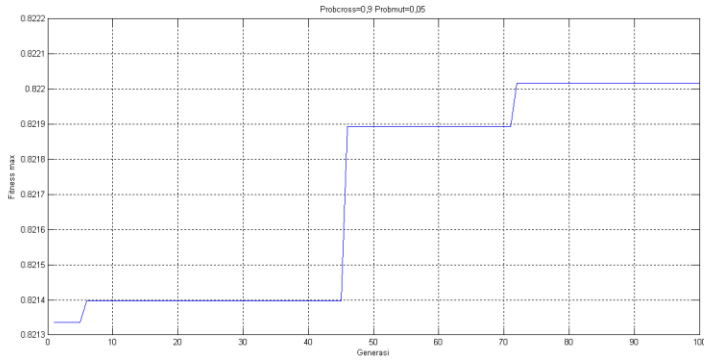


Gambar 4.19 Nilai *Fitness* Maksimal Probabilitas *Crossover*=0,01 dan Probabilitas Mutasi =0,2

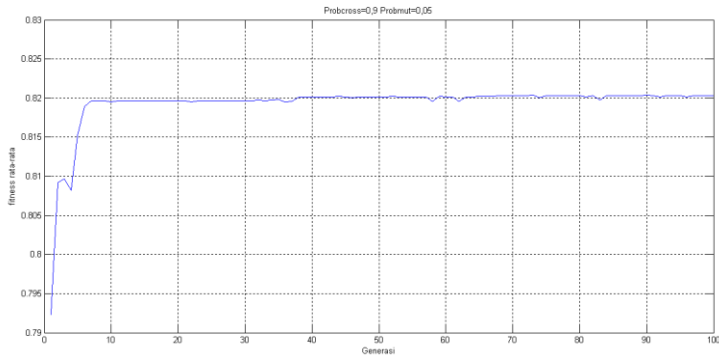


Gambar 4.20 Nilai *Fitness* Rata-rata Probabilitas *Crossover*=0,01 dan Probabilitas Mutasi =0,2

Hasil *tunning* yang didapatkan adalah $K_p=43,3138$, $K_i=0,1242$, dan $K_d=3,9922$ dengan nilai *fitness* maksimalnya adalah 0,8195. Kemudian dilakukan dengan menggunakan nilai Probabilitas *Crossover* 0,9 dan Probabilitas Mutasi 0,05. Grafik *fitness* yang didapatkan ditunjukkan pada Gambar 4.21

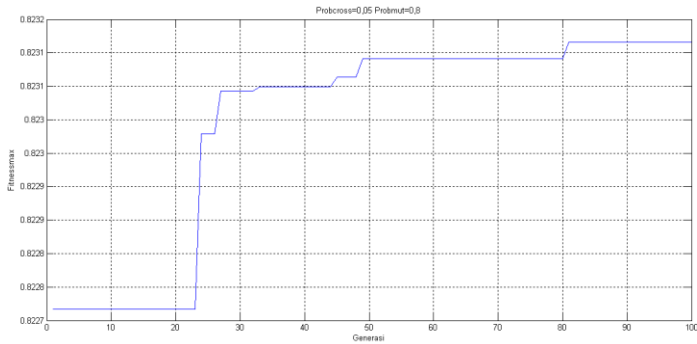


Gambar 4.21 Nilai *Fitness* Maksimal Probabilitas *Crossover*=0,3 dan Probabilitas Mutasi =0,02

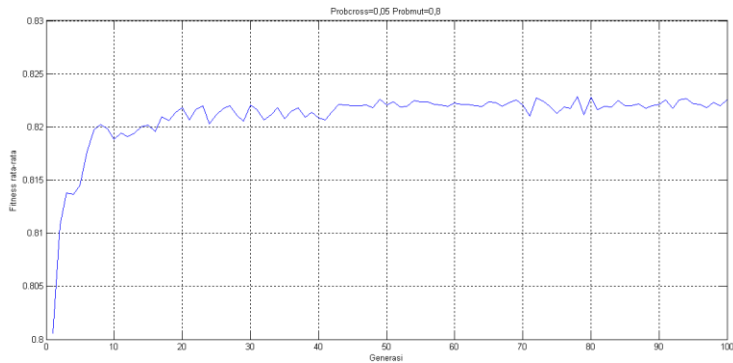


Gambar 4.22 Nilai *Fitness* Rata-rata Probabilitas *Crossover*=0,3 dan Probabilitas Mutasi =0,02

Hasil *tunning* yang didapatkan adalah $K_p=43,3138$, $K_i=0,1242$, dan $K_d=3,9922$ dengan nilai *fitness* maksimalnya adalah 0,8195. Kemudian dilakukan dengan menggunakan nilai Probabilitas *Crossover* 0,05 dan Probabilitas Mutasi 0,8. Grafik *fitness* yang didapatkan ditunjukkan pada Gambar 4.23



Gambar 4.23 Nilai *Fitness* Maksimal Probabilitas *Crossover*=0,05 dan Probabilitas Mutasi =0,8



Gambar 4.24 Nilai *Fitness* Rata-rata Probabilitas *Crossover*=0,05 dan Probabilitas Mutasi =0,8

Hasil *tunning* yang didapatkan adalah $K_p=49,29$, $K_i=0,1197$, dan $K_d=3,9454$. Dengan nilai *fitness* maksimalnya adalah 0,8192.

Dari 7 pengujian *learning* dengan menggunakan Algoritma Genetika didapatkan nilai maksimal *fitness* yang bisa didapatkan adalah sebesar 0,8231 pada pengujian dengan menggunakan Probabilitas *Crossover* dengan nilai 0,8 dan pengujian Probabilitas Mutasi 0,6. Sehingga nilai K_p , K_i , dan K_d yang digunakan pada simulasi adalah $K_p=50$, $K_i=0,1202$, dan $K_d=3,95$.

Dari Proses yang dilakukan dapat dilihat bahwa nilai *fitness* maksimal akan selalu naik. Hal ini dikarenakan terdapat penambahan

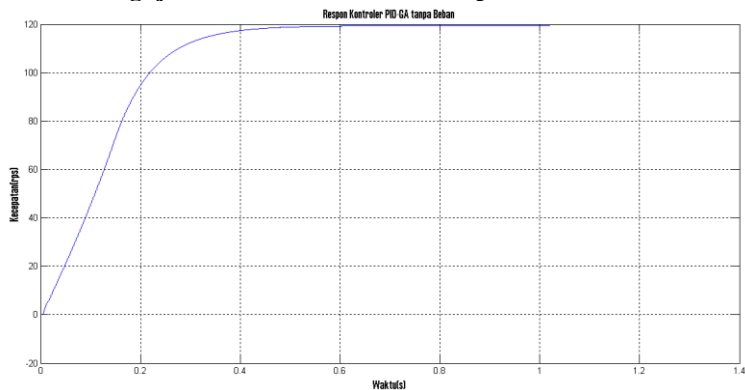
variabel *elitism* pada program pencarian individu terbaik. Sehingga individu terbaik pada generasi sekarang akan tetap digunakan sampai terdapat individu yang mempunyai nilai *fitness* yang lebih baik pada proses selanjutnya yang akan menggantikan.

Dari proses *learning* juga dapat dilihat bahwa nilai *fitness* rata-rata dari setiap generasi mempunyai fluktuasi nilai yang bervariasi dikarenakan selalu terdapat individu baru yang mempunyai nilai *fitness* yang baru pada setiap generasi.

4.4.2 Pengujian Kontroler PID-GA

Pada bagian ini dilakukan pengujian tanpa menggunakan beban dan menggunakan beban. Nilai parameter yang digunakan adalah nilai parameter hasil percobaan ke lima proses *learning* yaitu $K_p=50$, $K_i=0,1202$, dan $K_d=3,95$.

4.4.2.1 Pengujian Kontroler PID-GA Tanpa Beban

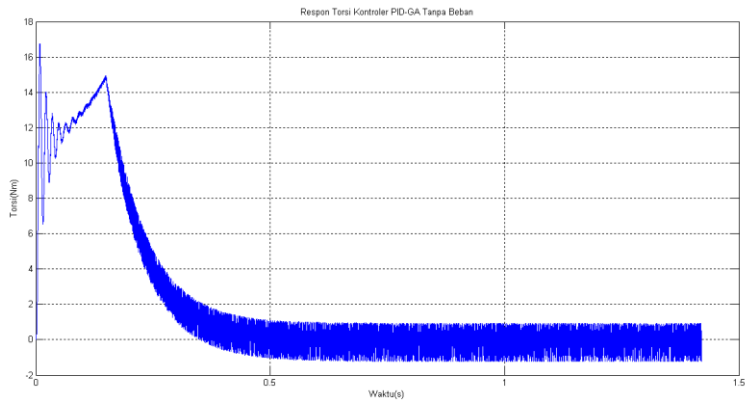


Gambar 4.25 Respon Kecepatan Kontroler PID-GA Tanpa Beban

Berdasarkan Gambar 4.25 di atas respon kecepatan motor induksi dengan DTC-PID berbasis Algoritma Genetika dapat dilihat bahwa tidak ada *overshoot* pada grafik respon, dan bentuk grafik respon seperti sistem orde pertama. Bentuk grafik respon yang seperti ini dikarenakan adanya penambahan kontroler PID pada sistem.

Respon kecepatan mencapai nilai referensi yang ditentukan yaitu 120 rps. Nilai konstanta waktu (τ) dari sistem merupakan waktu ketika sistem mencapai 63,2 % dari *steady state* (120/nilai referensi). Nilai

konstanta waktu (τ) untuk sistem mencapai kondisi kecepatan $0,632 \times 120 \text{ rps} = 75,84 \text{ rps}$. Sistem mencapai kondisi tersebut ketika $t = 0,32$ detik, sehingga nilai t inilah yang merupakan nilai konstanta waktu (τ) dari sistem. Dengan demikian diperoleh *settling time* sebesar $0,512$ detik. Respon torsi elektromagnetik dapat ditunjukkan pada Gambar 4.26.

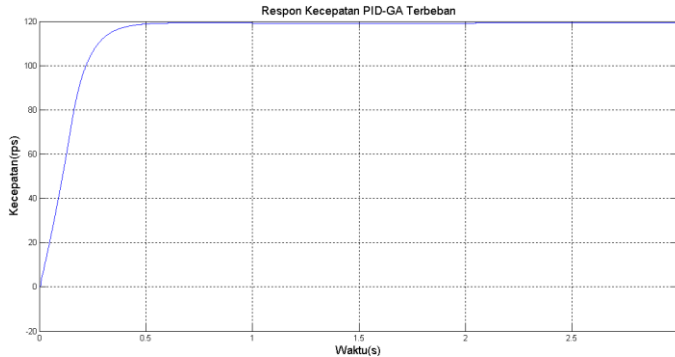


Gambar 4.26 Respon Torsi Kontroler PID-GA Tanpa Beban

Jika kita amati Gambar 4.26 di atas, hampir tidak ada perbedaan antara respon torsi elektromagnetik DTC-PID dan FLC-DTC dengan respon torsi elektromagnetik DTC-PID berbasis GA pada motor induksi. Namun dengan digunakan metode DTC-PID berbasis Algoritma Genetika, fluktuasi *ripple* torsi yang dibangkitkan pada saat *steady state* sebesar $-0,98 \text{ Nm}$ sampai dengan $0,78 \text{ Nm}$. Hal ini membuktikan bahwa fluktuasi *ripple* torsi lebih kecil jika dibandingkan dengan DTC-PID dan FLC-DTC. Dengan kata lain bahwa fluktuasi *ripple* torsi yang dibangkitkan pada saat *steady state* sebesar $1,76 \text{ Nm}$ jika sebelumnya $2,201 \text{ Nm}$. Nilai ini didapatkan dengan cara menghitung jarak antara nilai *ripple* torsi terendah ke nilai *ripple* torsi tertinggi. Dan respon torsi pada saat *rise time* terdapat *overshoot* sebesar $16,25 \text{ Nm}$ pada waktu $0,01946$ detik.

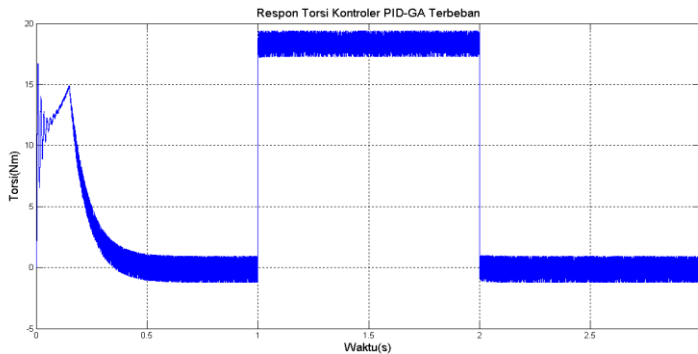
4.4.2.2 Pengujian Kontroler PID-GA Terbeban

Gambar 4.25 menunjukkan respon kecepatan dan torsi motor induksi ketika diberi beban sebesar $22,5 \text{ Nm}$, diterapkan pada 1 detik dan beban dihilangkan pada 2 detik



Gambar 4.27 Respon Kecepatan Kontroler PID-GA Terbeban

Dengan menggunakan kontroler PID-GA didapatkan penurunan kecepatan yang lebih kecil dengan referensi kecepatan 120 rps yang turun menjadi 119,2 rps. Hal ini menunjukkan hasil *tunning* PID menggunakan GA mampu meminimalisir *error* yang terjadi pada respon kecepatan



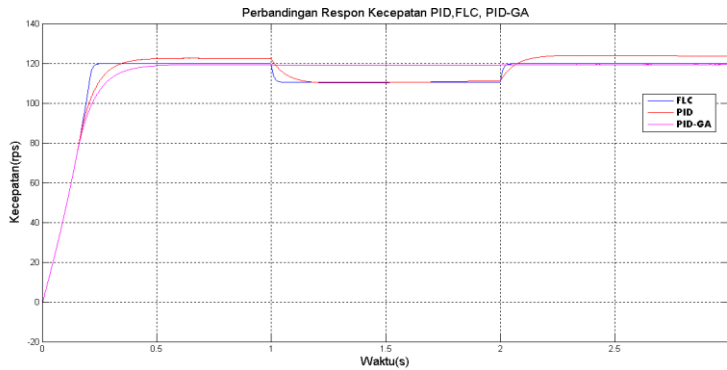
Gambar 4.28 Respon Torsi Kontroler PID-GA Terbeban

Respon torsi yang ditunjukkan pada Gambar 4.28 jika diamati terdapat perbedaan yaitu pada fluktuasi *ripple* minimum.

4.4.2.3 Perbandingan Kontroler PID-GA, PID, dan FLC

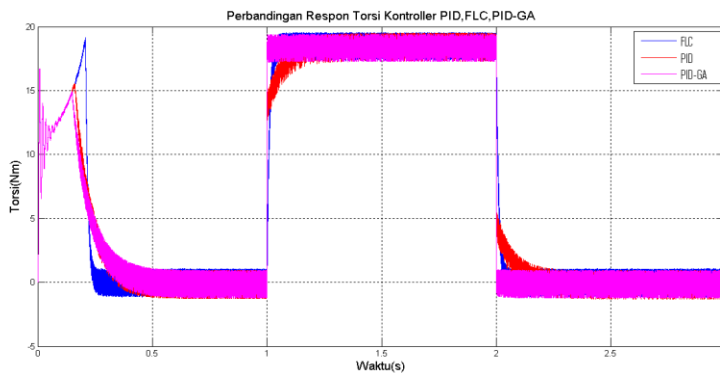
Untuk bisa lebih jelas mengamati perbedaan hasil dari perancangan Kontroler PID, FLC, dan PID berbasis GA maka perlu

ditunjukkan *plot* perbandingan grafik dari ketiganya. Gambar 4.29 menunjukkan hasil perbandingan Kecepatan antara kontroler PID, FLC, dan PID-GA yang diberikan beban nominal



Gambar 4.29 Perbandingan Respon Kecepatan PID, FLC dan PID-GA

Dari hasil perbandingan dapat dilihat bahwa kontroler PID-GA mampu mengatasi penurunan kecepatan dengan baik dibandingkan dengan kontroler PID dan FLC. Namun untuk mencapai keadaan *steady state* respon PID-GA membutuhkan waktu yang lebih lama dibandingkan dengan menggunakan PID dan FLC. Kemudian juga dibandingkan respon torsi dari keduanya yang ditunjukkan pada Gambar 4.30



Gambar 4.30 Perbandingan Respon Torsi PID, FLC dan PID-GA

Dari Gambar 4.27 dapat diketahui bahwa respon torsi dari PID-GA mampu lebih cepat mencapai *steady state* dibandingkan dengan respon torsi PID dan FLC. Perbandingan kontroler PID, FLC dan PID-GA dilihat dari respon waktu disajikan pada Tabel 4.2

Tabel 4.2 Perbandingan Karakteristik Respon PID dan PID-GA

Kecepatan (rps)	Karakteristik respon	PID (detik)	FLC (detik)	PID-GA (detik)
120	τ	0,126	0,122	0,322
	t_s	0,4201	0,225	0,5121
	t_r	0,2814	0,25	0,3876
	t_d	0,0912	0,074	0,107

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil perancangan, simulasi dan analisis hasil simulasi maka dapat diperoleh kesimpulan sebagai berikut:

1. Proses pengontrolan model motor induksi DTC dengan kontroler PID berbasis Algoritma Genetika dapat memberikan respon kecepatan rotor yang membentuk grafik respon seperti karakteristik sistem orde pertama. Selain itu grafik respon mampu mencapai nilai kecepatan acuan yang diberikan yaitu 120 rps dengan rata-rata *settling time* yaitu 0,512 detik.
2. Desain DTC-PID berbasis Algoritma Genetika dapat meminimalkan fluktuasi *ripple* torsi pada saat *steady state* yaitu sebesar 1,78 Nm jika dibandingkan dengan DTC-PID yang mempunyai fluktuasi *ripple* torsi yang dibangkitkan Dengan digunakan kontroler DTC-PID, DTC-FLC, dan DTC-PID berbasis Algoritma Genetika jika dibandingkan terdapat perbedaan yang cukup signifikan dalam penurunan kecepatan saat terbeban dimana saat diberikan beban 18,5 Nm penurunan kecepatan yang terjadi pada penggunaan Kontroler PID dan FLC sebesar 10 rps sedangkan menggunakan kontroler PID-GA sebesar 0,8 rps dan didapatkan *ripple* torsi sebesar 2,201 Nm.

5.2 Saran

Dari hasil penelitian yang telah dilakukan, maka penulis menyarankan untuk penelitian ke depannya:

1. Karena penelitian ini hanya bersifat simulasi dari model matematis motor induksi dan komponen peralatan lainnya, maka penelitian ke depan perlu diimplementasikan dan diujicobakan langsung pada peralatan real. Untuk mendapatkan performa yang lebih baik lagi sistem pengaturan kecepatan motor induksi dengan secara *online* sehingga tidak perlu dilakukan proses pemodelan
2. Perlu pengkajian yang lebih mendalam dalam menerapkan Algoritma Genetika dalam optimasi parameter PID supaya mendapatkan hasil yang lebih baik lagi

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Popescu, M., "*Induction Motor Modelling for Vector Control Purposes*," Finland: Helsinki University of Technology, 2000
- [2] Rameli, Mochammad. "*Bahan Kuliah Pengaturan Penggerak Elektrik*." Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Institute Teknologi Sepuluh Nopember, Surabaya.
- [3] Yusuf, Y.I., Pengaturan Kecepatan Motor Induksi Tiga Fasa Dengan Direct Torque Control Berbasis Neuro Fuzzy. *Tugas Akhir*. Teknik Elektro. Surabaya: Institut Teknologi Sepuluh Nopember. 2015
- [4] R., Krishnan. *Electric Motor Drives Modelling ,Analysis, and Control*. , .New York: Prentice Hall. 2001
- [5] Amandeep Jakhar and Prerna Gaur, "Comparative Analysis of GA Tuned PID Controller for Direct Torque Control," in *IEEE International Conference on Computer, Communication and Control (IC4-2015)*, Indore: India, 2015
- [6] Ogata, Katsuhiko. *Modern Control Engineering*. New Jersey : Prentice Hall. 1997.
- [7] Chipperfield, "Introduction to genetic algorithms," in Genetic Algorithms in Engineering Systems, Stevenage, *IEE Engineering Series*, 1997, pp. 1-41.
- [8] Obitko, *Introduction to Genetic Algorithm*. [Online] Available at: <http://www.obitko.com/tutorials/geneticalgorithms/index.php>. 1998 [Accessed at 21 April 2017]
- [9] Ying Chen, Yong-jie Ma, and Wen-xia Yun, "Application of Improved Genetic Algorithm in PID Controller Parameters Optimization," *Telkomnika*, pp. 1524-1530, 2013

- [10] Uddin M.Nasir and Muhammad Hafeez,” FLC Based DTC Scheme to Improve the Dynamic Performance of an IM Drive” *IEEE Trans. Ind. Appl.*, vol. 48, no. 2, pp. 823-831.2011.
- [11] Kurniawan, Marendra., Perancangan dan Implementasi Pengaturan Kecepatan Motor Tiga Fasa pada Mesin Sentrifugal Menggunakan Kontroler Logika Fuzzy. *Tugas Akhir*. Teknik Elektro. Surabaya: Institut Teknologi Sepuluh Nopember. 2014

LAMPIRAN

LAMPIRAN 1

```
    ukkro          = 30;
    uksol          = 3;
    ukbit          = ukkro/uksol;
    ukpop          = 30;
    minKp          = 30%30;
    maxKp          = 50%50;%80
    minTi          = 0.01;
    maxTi          = 0.15%6000
    minTd          = 0.01%0.01;
    maxTd          = 4%4;%150

    pop            = rand(30,30)<0.5;
    popbin         = double(pop);
    genmax         = 40;
    probcross      = 0.8%0.01;%0.8%0.1%0.3%0.9%0.5%0.8
    probmut        = 0.6%0.2;%0.02%0.1%0.02%0.05%0.05%0.6
    elit           = 1;

    gen=0;

while gen<genmax
    for f=1:ukpop
        kpdec=polyval(pop(f,1:10),2)
        kidec=polyval(pop(f,11:20),2)
        kddec=polyval(pop(f,21:30),2)

1)) *kpdec;      out1          = minKp+((maxKp-minKp)/((2^ukbit)-
1)) *kidec;      out2          = minTi+((maxTi-minTi)/((2^ukbit)-
1)) *kddec;      out3          = minTd+((maxTd-minTd)/((2^ukbit)-
1)) *kddec;      Kp            = out1;
                  Ki            = out2;
                  Kd            = out3;

        cont = Kp + Ki/s + Kd*s;

        dt=0.01;
        t=0:dt:3;
        sim('cobalangsungpidtc');
        err=keluar;
        J=sqrt(mean(abs(err)))
        fitness(f,1) = 1/(1+J);
    end

    fitrat(gen+1,1)          = mean(fitness);
```

```

        fitmax(gen+1,1)                = max(fitness);
        [fitness,indfit]                =
sort(fitness,'descend');
        popurut                        = pop(indfit,:);

        if (gen == genmax)
            break
        end
        pop                            = popurut;
%-----Tournament Selection-----
%-----%
%-----%
%-----%

        for(j=elit+1:ukpop)

            for(i=1:3)
                indtor                = randi([2 ukpop],1);
                ortu(i,:)              = pop(indtor,:);
                fittor(i,1)            = fitness(indtor,1);
            end
            [fittor,indfit]            = max(fittor);
            poptor(j,:)               = ortu(indfit,:);
            pop(j,1:ukkro)            = poptor(j,1:ukkro);

        end

%-----Crossover-----
%-----%
%-----%

        for(j=elit+1:ukpop)

            randcross                  = rand(1,1);
            if randcross<probcross
                for(i=1:2)
                    indcross(i,1)      = randi([2 30]);
                    cross(i,1:ukkro)   = pop(indcross(i,1),1:ukkro);
                end
                pointcross1= randi(ukkro);
                pointcross2= randi(ukkro);
                if pointcross1>pointcross2

                    cross(2,pointcross1:pointcross2)=cross(1,pointcross1:pointcross2);
                else
                    cross(2,pointcross2:pointcross1)=cross(1,pointcross2:pointcross1);
                end
                popcross(j,1:ukkro)=cross(2,1:ukkro);
            end
        end
    end
end

```



```

        pop(j,1:ukkro)=popcross(j,1:ukkro);
    else
        pop(j,1:ukkro)      = pop(j,1:ukkro);
    end

end

%-----Mutasi-----
%-----%
%-----%
%-----%
    for(j=elit+1:ukpop)

        randmut          = rand(1,1);
        if randmut<probmud
            mutkol          = randi(ukkro);
            popmut(j,1:mutkol-1) = pop(j,1:mutkol-1);
            popmut(j,mutkol)      = not(pop(j,mutkol));
            popmut(j,mutkol+1:ukkro)= pop(j,mutkol+1:ukkro);
        else
            popmut(j,1:ukkro)=pop(j,1:ukkro);
        end
        pop(j,1:ukkro)=popmut(j,1:ukkro);
    end

    %   popbaru          = pop;
    %   pop(elit+1:ukpop,:) = popbaru(elit+1:ukpop,:);
    gen=gen+1;
    generasi(gen)=gen;
end

```


RIWAYAT HIDUP PENULIS



Adeta Gita Ahmadani, dilahirkan di Batam pada tanggal 24 Mei 1994. Merupakan Anak Pertama Bapak Eko Purwanto dan Ibu Sri Winarni Menempuh jenjang pendidikan SD Negeri 11 Ngringo (1999-2005), SMP N 2 Cikarang Barat (2005-2008), SMA N 1 Cikarang Utara (2008-2011). Setelah lulus dari SMA, penulis melanjutkan pendidikan di Diploma 3 Teknik Elektro Sekolah Vokasi Universitas Gadjah Mada dan mengambil bidang Arus Lemah. Setelah lulus pada tahun 2014, penulis melanjutkan pendidikan program Sarjana di Institut Teknologi Sepuluh Nopember Surabaya (ITS Surabaya) pada tahun 2015 melalui jalur penerimaan lintas jalur dan mengambil bidang Teknik Sistem Pengaturan.

email : adetaahmadani@gmail.com